WEBDYN

MANUEL D'UTILISATION

WebdynRF

Concentrateur dédié au smart metering et à la maîtrise des énergies

SOMMAIRE

1 Remarques concernant ce manuel 4
1.1 Champ d'application 4
1.2 Versions de produits 4
1.3 Groupe cible 5
2 Principe général de fonctionnement 5
3 Communication avec le serveur distant 6
3.1 Modes de connexion 6
3.1.1 Gestion du code PIN de la carte SIM 7
3.2 Serveur FTP distant 7
3.3 Serveur Web Services distant 8
3.4 Connexion au serveur distant 9
3.4.1 Process de téléchargement sur le serveur distant (Upload) 10
3.4.2 Inbox 12
3.5 Bouton Request 12
4 Commandes 13
4.1 Commandes Wavenis : Scan, TimeSync 13
4.2 Commandes spécifiques Wavenis 15
4.3 Commandes spécifiques Modbus 17
4.4 Commande Status 17
4.5 Commande de mise à jour 18
4.6 Commande d'un contact sec (sortie numérique) 18
5 Configuration de la passerelle 18
5.1 Paramètres 18
5.2 Configuration par SMS 21
5.3 Configuration locale 23
5.4 Configuration distante 23
5.5 Contrôle de l'accès local 24
5.6 Configuration des ports 25
6 Wavenis 26
6.1 Mode transparent 26
6.2 Modules Wavenis supportés 27
6.3 Configuration Wavenis 27
6.4 Alarmes Wavenis 28
6.5 La gestion du temps Wavenis 29
6.6 Support des modules WaveTIC 30
6.7 Erreurs de communication Wavenis 30
7 Impulsions 31
8 Wired M-Bus 31
9 Wireless M-Bus 31
10 Tags RFID actifs 32
11 Modbus 33
11.1 Configuration 34
11.2 Ensembles de données Modbus 34
11.2.1 Variables 35

11.2.2 Boundaries 37
11.3 Esclaves Modbus 37
11.4 Adresse des variables 37
12 Scheduler 38
13 Moteur d'alarme 41
14 Fichiers de log 43
15 Synchronisation de l'horloge interne 45
16 Mise à jour du firmware de la passerelle 45
17 Support 46
18 ANNEXE A Schéma XSD – Configuration 47
19 ANNEXE B Exemple XML – Configuration 69
20 ANNEXE C Schéma XSD – Alarmes 75
21 ANNEXE D Exemple XML – Alarmes 87
22 ANNEXE E Schéma XSD – Supervision 93
23 ANNEXE F Exemple XML – Supervision 99
24 ANNEXE G Schéma XSD – Données 102
25 ANNEXE H Format CSV – Données 109
26 ANNEXE I Schéma XSD – Commandes 110
27 ANNEXE J Exemple XML – Commandes 124

1 Remarques concernant ce manuel

Ce guide décrit le montage, l'installation et la mise en service de l'appareil. Veuillez garder ces instructions toujours disponibles.
Pour les instructions de gestion des données échangées par la passerelle avec le serveur distant se référer au manuel d'exploitation de la passerelle WebdynRF (cf. : ME-WebdynRF_FR.pdf).

1.1 Champ d'application

La présente description technique est valable pour les passerelles WebdynRF-Wavenis à partir de la version de matériel 1 et de la version logicielle 2.4.0.

1.2 Versions de produits

Suivant le type de modem GSM et de carte radio, il existe plusieurs versions de produit :

Carte radio Wavenis :

Références produits	Versions
WG0606-A01	Modem 2G - Wavenis 868MHz/25mW
WG0606-A02	Modem 3G - Wavenis 868MHz/25mW
WG0606-A03	Modem 2G - Wavenis 868MHz/500mW
WG0606-A04	Modem 3G - Wavenis 868MHz/500mW
WG0606-A11	Modem 2G - Wavenis 915MHz/25mW
WG0606-A12	Modem 3G - Wavenis 915MHz/25mW
WG0606-A13	Modem 2G - Wavenis 915MHz/500mW
WG0606-A14	Modem 3G - Wavenis 915MHz/500mW
WG0606-A21	Modem 2G - Wavenis 433MHz/10mW
WG0606-A22	Modem 3G - Wavenis 433MHz/10mW

Carte radio Wireless M-Bus :

Références produits	Versions
WG0607-A01	WebdynRF-WirelessMbus-868MHz-2G
WG0607-A02	WebdynRF-WirelessMbus-868MHz-3G
WG0607-A11	WebdynRF-WirelessMbus-169MHz-2G
WG0607-A12	WebdynRF-WirelessMbus-169MHz-3G

Carte radio RFID :

Références produits	Versions
WG0608-A01	Modem 2G - RFID

Accessoires commun à tous les produits :

Références produits	Versions
AC0102-02	Bloc Alimentation externe 12v
AC0103-00	Alimentation 24V Rail Din
AC0201-01	Antenne GPRS bi bandes déportée avec 5m de câble
AC0201-02	Antenne GPRS bi bandes déportée avec 20m de câble
AC0201-03	Antenne GPRS bi bandes déportée avec 10m de câble
AC0501-01	Boîtier IP67, Disjoncteur, alimentation 24V DC, bloc UPS et batteries au plomb 12V

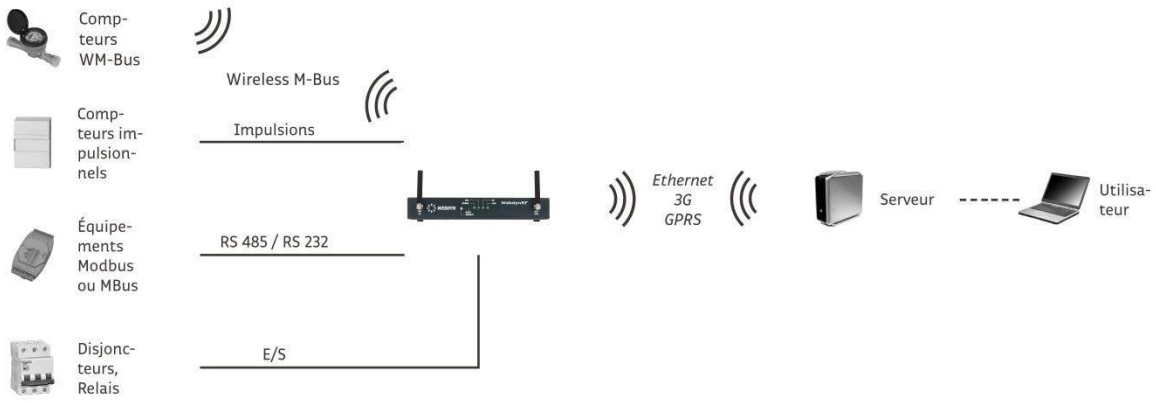
Ce manuel s'adresse à toutes les versions de produit.

1.3 Groupe cible

Ce guide s'adresse aux utilisateurs/exploitants de la passerelle, pour leur permettre de traiter les données remontées par la passerelle et la configurer à distance.

2 Principe général de fonctionnement

La plate-forme WebdynRF est la nouvelle gamme de concentrateurs Webdyn dédiée aux réseaux sans fil. Collecter des données de compteurs ou de capteurs, contrôler des E/S sont les fonctionnalités de base de la passerelle WebdynRF. Les services visés par la passerelle WebdynRF sont la télé relève, ou la maîtrise de l'énergie dans le bâtiment.



Spécifications Techniques : WebdynRF Wireless M-Bus

TECHNIQUES

- **Interface radio :** 868/169MHz
- **Interfaces E/S :** 3 entrées numériques TOR - impulsions et 1 sortie numérique - relais
- **Interfaces série :** 1 port RS485 Modbus et 1 port RS232 Modbus/M-Bus
- **Interfaces réseau IP :** ethernet : 10/100Mbits/s – GSM/GPRS : 850/900/1800/1900Mhz

GÉNÉRALES

- **Alimentation :** batterie lithium-ion [+12/24v] DC fourni par une alimentation externe (Accessoire)
- **Batterie interne :** Li-Ion 650mAH
- **Batterie externe :** gestion d'un UPS externe par trois entrées TOR dédiées
- **T° :** fonctionnement : -20°C/+70°C - Stockage : -20°C/+85°C
- **Dimensions :** boitier métallique compact de 20x12x3,2cm
- **Agrément RTTE :** 1995/5/CEEE ; EN 13757-4 (OMS)
- **Consommation :** 7W maximum

SUPERVISION

- Supervision d'un réseau de capteurs Wireless M-Bus suivant les modes S1 et T1 dans la bande 868MHz ou suivant le mode N dans la bande 169MHz
- Supervision d'un port série RS232 (Modbus ou M-Bus)
- Supervision d'un port série RS485 Modbus
- Supervision de trois entrées TOR (numérique ou impulsions)
- Supervision d'une sortie relais

3 Communication avec le serveur distant

La passerelle communique avec un serveur distant via un serveur FTP et/ou un serveur Web Services. ***Gestion du serveur Web Services non-implémenté***

Chaque fois que sa configuration est modifiée, la passerelle peut soit la télécharger sur un serveur FTP soit l'envoyer vers un serveur Web Services (WS).
De même, les alarmes et les données collectées peuvent être téléchargés sur un serveur FTP ou envoyées à un serveur Web Services.

Lorsque vous utilisez le transfert FTP, la passerelle peut également informer un serveur Web Services de tout téléchargement FTP.

Le serveur peut aussi lancer des actions sur la passerelle en plaçant des fichiers de commande dans un répertoire INBOX sur le serveur FTP ou en lui envoyant des commandes lorsque la passerelle interroge le Web Services INBOX. Les commandes peuvent également être envoyées à la passerelle par SMS.

3.1 Modes de connexion

La connexion au serveur distant peut être établie via Ethernet ou un réseau cellulaire (GPRS ou 3G en fonction de la configuration matérielle). Les échanges entre la passerelle et le serveur distant sont toujours initiés par la passerelle, mais différentes méthodes sont disponibles pour le serveur distant pour déclencher un échange (voir paragraphe 3.4).
La passerelle peut être configurée pour utiliser le modem dans un des quatre modes suivants :

On demand : Dans ce mode, le lien PPP est créé lorsque la passerelle a besoin de communiquer avec le serveur distant. Le lien sera fermé après l'achèvement de la communication avec le serveur distant.

Always On : Dans ce mode, le lien PPP sera maintenue en permanence de façon indépendante. Dans ce mode, un mécanisme de keepalive peut être activé pour s'assurer que le lien est fonctionnel. ***Mode Always On non-implémenté***

Always Off : Dans ce mode, le lien PPP n'est jamais créé. Toutes les communications avec le serveur passent par l'interface Ethernet. Le modem (si une carte SIM valide est présente) est cependant connecté au réseau cellulaire, prêt à recevoir les appels entrants et / ou SMS.

Off : Dans ce mode, le modem est hors tension.

3.1.1 Gestion du code PIN de la carte SIM

La passerelle peut être configurée pour utiliser une carte SIM:

- Sans code PIN : `/com/modem/pin/mode=off`
- Avec le code PIN : `/com/modem/pin/mode=manual` et `/com/modem/pin/code=<CODE PIN>`
- Avec un code PIN automatique : ***Mode automatique non-implémenté***

3.2 Serveur FTP distant

La passerelle utilise les fichiers suivants sur le serveur FTP :

Nom	Description
CONFIG/<uid>.xml	Configuration actuelle de la passerelle. Ce fichier est transféré par la passerelle après chaque modification de sa configuration. La modification de ce fichier n'a pas d'effet sur la passerelle. La passerelle va simplement l'écraser la prochaine fois que sa configuration est modifiée (voir ci-dessous INBOX).
DATA/<uid>-<timestamp>.<format>.gz	Fichiers de données téléchargés par la passerelle sur le serveur distant.
ALARM/<uid>-<timestamp>.xml.gz	Fichiers d'alarme téléchargés par la passerelle sur le serveur distant.
SUPERVISION/<uid>-<timestamp>.xml.gz	Fichiers de supervision téléchargés par la passerelle sur le serveur distant (états et résultats de scan).
SUPERVISION/<uid>-<timestamp>.log.gz	Fichiers de logs téléchargés par la passerelle sur le serveur distant sur requête.
INBOX/<uid>/*.xml	La passerelle surveille ce répertoire. Tout fichier placé dans celui-ci sera téléchargé et traité par la passerelle.
BIN/<firmware>	Ce répertoire contient le firmware de la passerelle pour réaliser une mise à jour

Dans le tableau ci-dessus, `<uid>`, `<timestamp>` et `<format>` doivent être remplacés respectivement par l'identifiant unique de la passerelle, l'horodatage du téléchargement et le format sélectionné (csv ou xml).

Le format d'horodatage est «AAAAMMJJ-HHMMSS» de sorte qu'un tri alphabétique du répertoire donne l'ordre chronologique.
Les fichiers avec l'extension « .gz » sont compressés.

La passerelle télécharge toujours les fichiers en suivant un processus en 2 étapes :

- Le fichier est téléchargé avec une extension supplémentaire « .tmp ».
- Le fichier est renommé en supprimant l'extension « .tmp ».

Ce processus permet au serveur distant de distinguer facilement les fichiers en cours de téléchargement des fichiers complètement téléchargés.

Le schéma XML de configuration est présenté dans le paragraphe 18.

Un exemple de fichier XML de configuration est présenté dans le paragraphe 19.

Le schéma XML des alarmes est présenté dans le paragraphe 20.
Un exemple de fichier XML d'alarmes est présenté dans le paragraphe 21.

Le schéma XML de supervision est présenté dans le paragraphe 22.
Un exemple de fichier XML de supervision est présenté dans le paragraphe 23.

Le format CSV des données est présenté dans le paragraphe 0.

Le schéma XML des commandes est présenté dans le paragraphe 26.
Un exemple de fichier XML de commande est présenté dans le paragraphe 27.

Les schémas XML spécifiant le format des différents fichiers XML utilisés par la passerelle peut évoluer dans les futures versions lorsque de nouvelles fonctionnalités seront ajoutées. Ces changements seront apportés afin que les anciens fichiers XML restent compatibles avec les nouveaux schémas XML. De même, comme les fichiers XML générés par la passerelle peuvent contenir des éléments supplémentaires, leur traitement doit être mis en œuvre afin que les nouveaux éléments soient ignorés.

3.3 Serveur Web Services distant

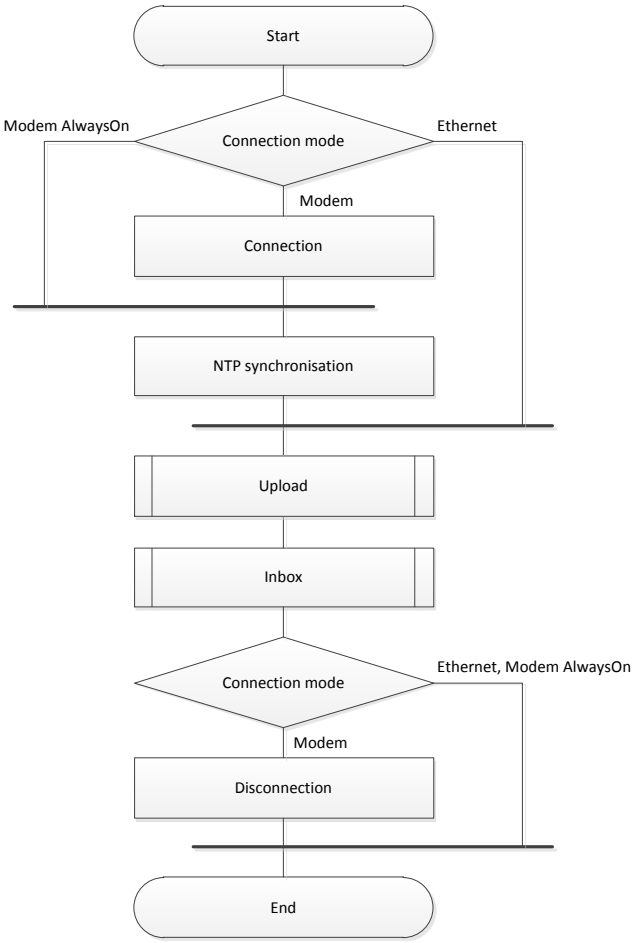
Non-implémenté

3.4 Connexion au serveur distant

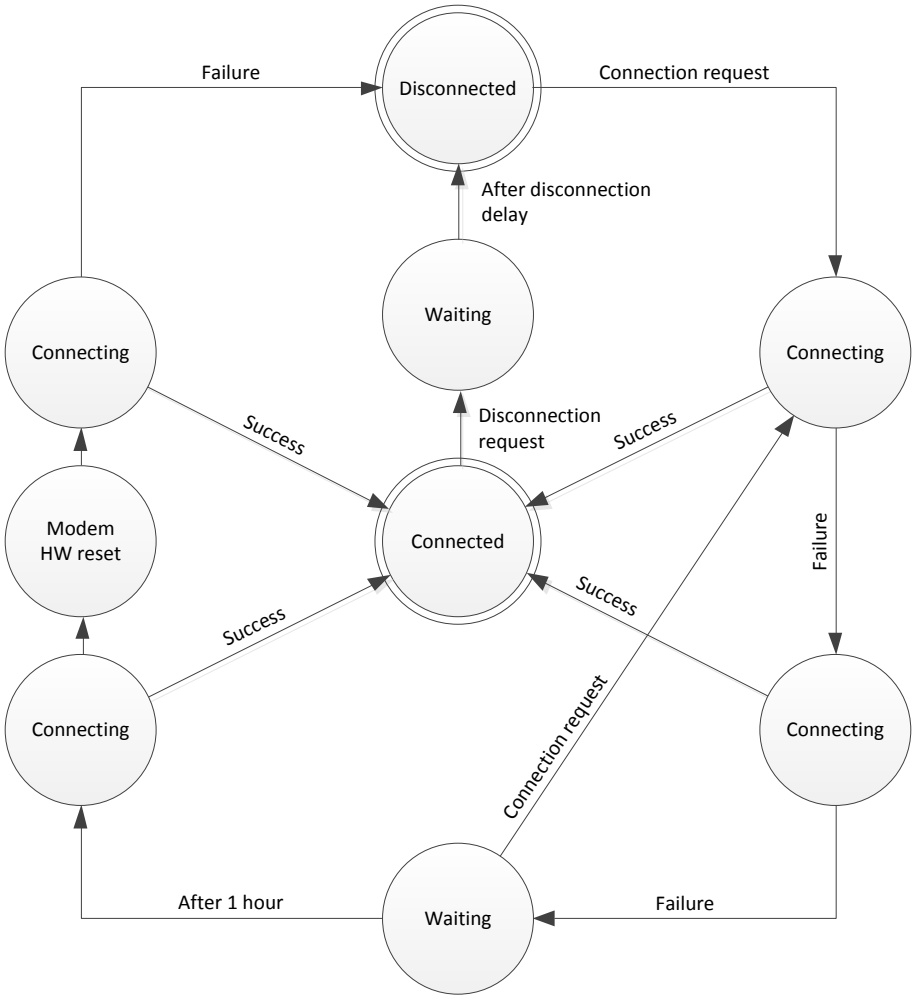
Une connexion au serveur distant peut être initiée par l'un des événements suivants :

- Scheduler de connexion
- Emission d'une alarme
- Modification de la configuration
- SMS de demande de connexion
- Bouton REQUEST (via les pages web ou le bouton en face avant du boîtier)

Indépendamment de l'événement déclencheur, le processus suivant est exécuté :



En cas d'échec de la connexion GPRS, une nouvelle connexion va être tentée une heure plus tard, à moins qu'une nouvelle connexion n'ait été lancée dans l'intervalle (sur demande explicite, ou déclenchée par une alarme ou par un envoi périodique). Ceci est illustré ci-dessous, précisant le délai de déconnexion :



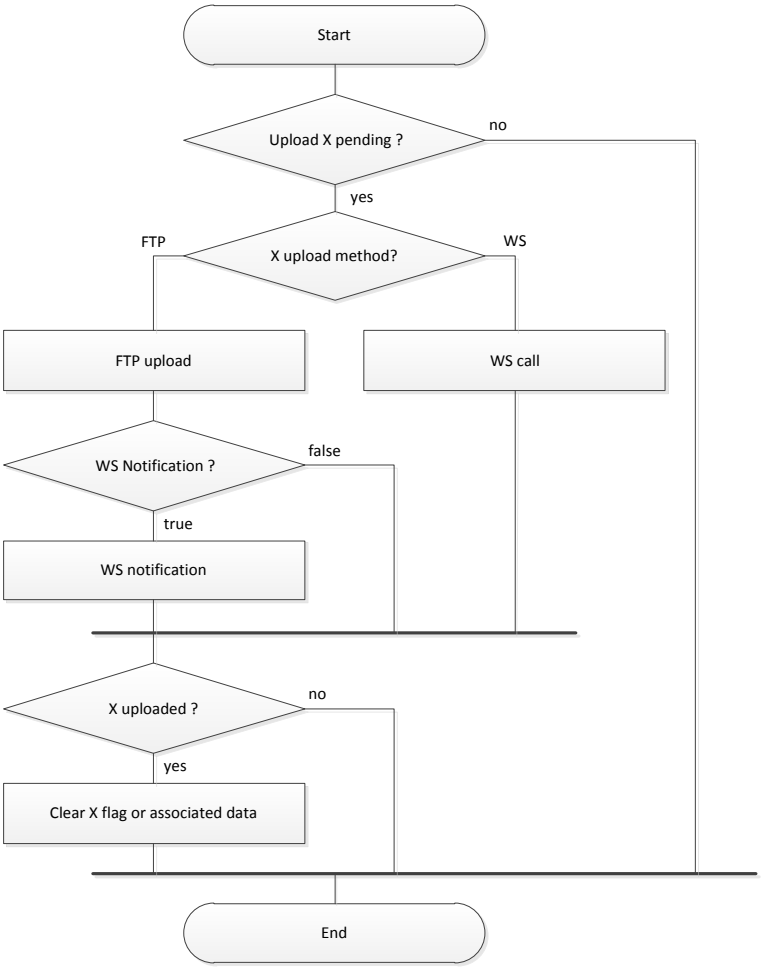
3.4.1 Process de téléchargement sur le serveur distant (Upload)

Configuration, alarme, supervision et données sont téléchargées sur le serveur distant par la passerelle de façon indépendante comme décrit dans le schéma ci-dessous.

Dans ce diagramme, X représente le type de téléchargement (configuration, alarme, supervision ou données).

Après l'achèvement d'un téléchargement d'une configuration, le drapeau associé est effacé.

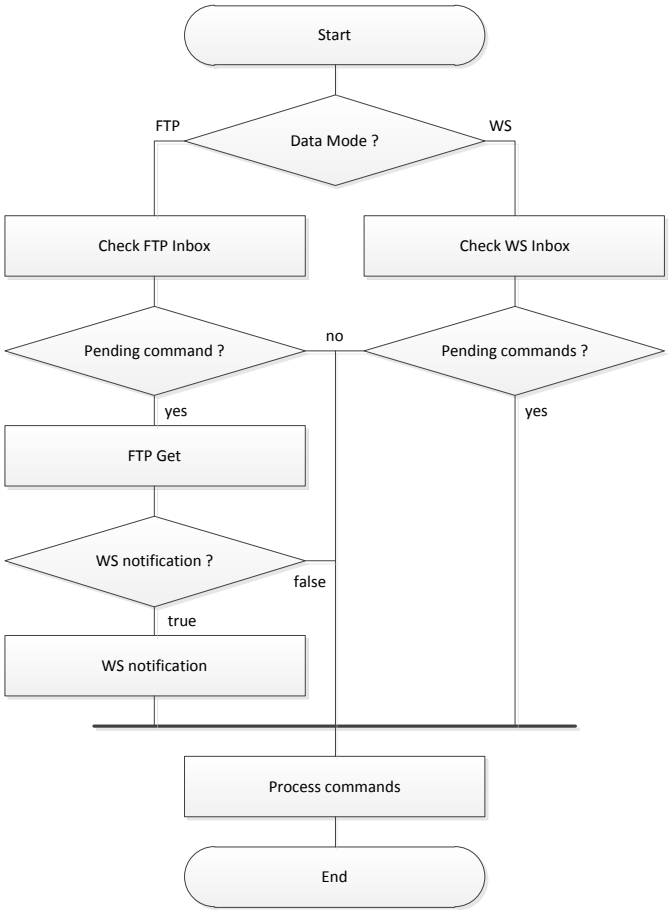
Après l'achèvement d'un téléchargement d'une alarme/supervision/données, les données associées sont effacées.



Remarque : La gestion du mode WS n'est pas implémentée

3.4.2 Inbox

La passerelle vérifie les actions en attente comme suit:



3.5 Bouton Request

Par défaut, appuyer sur le bouton «request» va déclencher une connexion au serveur distant et le téléchargement, en plus des données en attente, du status de la passerelle. Les deux peuvent être sélectivement désactivés avec les paramètres de configuration :
`/com/request/upload`
et
`/com/request/include_status`.

Un SMS d'état peut-être également envoyé au destinataire spécifié le paramètre suivant :
`/com/request/sms_status_recipient`
Dans le cas où ce champ est vide, aucun SMS ne sera envoyé.

4 Commandes

Les commandes peuvent être envoyées via le serveur distant (FTP ou WS) ou via SMS.
Lorsque la passerelle reçoit un SMS, elle vérifie la liste blanche des numéros de téléphone autorisés (caller_id). S'il est autorisé, le contenu du SMS est traité.

Command	Description	Retour
reboot	Redémarre la passerelle	Aucun
factory	Restaure les paramètres usine de la passerelle	Aucun
update	Lance la mise à jour du firmware de la passerelle	Alarme (SW)
scan	Lance un scan des valeurs instantanées de modules Wavenis, de leurs niveaux RSSI, de leurs niveaux de batterie, et/ou de leurs horloges RTC	Données Supervision (sauf données scan)
timesync	Lance la mise à jour des horloges RTC de modules Wavenis	Données Supervision
wavenis	Commande spécifique Wavenis (voir ci-dessous)	Alarme
status	Demande d'envoi du status de la passerelle	Données Supervision
log	Demande d'envoi des données de Log	Données Log
config	Modification de la configuration de la passerelle (seulement SMS)	Dépôt de la configuration
connect	Déclenche une connexion au serveur distant (seulement SMS)	Implicite (connexion)

Les commandes ne sont pas acquittées lorsqu'elles sont reçues. Toutes les commandes sont enregistrées et une commande non valide déclenche une alarme, qui est téléchargée vers le serveur distant.
Toutes les commandes acceptent deux paramètres facultatifs "uid" et "cid":

- `uid` : identifiant unique de la passerelle
- `cid` : identifiant de commande

Une commande sera rejetée si le paramètre `uid` inclus et ne correspond pas à l'`uid` de la passerelle.
Le `cid` peut être librement choisi par l'émetteur de la commande. Il sera inclus avec tout téléchargement associé.

Le schéma XML des commandes est présenté dans le paragraphe 26.
Un exemple de fichier XML de commande est présenté dans le paragraphe 27.

4.1 Commandes Wavenis : Scan, TimeSync

Une liste de modules Wavenis peut être spécifiée pour les commandes «`Scan`» et «`TimeSync`». Sinon, la commande est appliquée à tous les modules connus. Les commandes peuvent seulement être appliquées à des modules connus (toutes les adresses inconnues sont ignorées).

Les données demandées (sauf pour les données d'index immédiat) seront téléchargées sur le serveur distant en tant que données de supervision utilisant la méthode de téléchargement configurée (`/upload/supervision/method`). Les données d'index immédiat (données scan) seront téléchargées sur le serveur distant en tant que données Wavenis en utilisant la méthode de téléchargement configurée (`/upload/data/method`).

Exemple : Demande de mise à jour de l'horloge RTC de deux modules Wavenis :

```
XML:
<cmd cid='C_1234'>
   <timesync>
      <address>011A0A30AAA0</address>
      <address>011A0A30AAA1</address>
   </timesync>
</cmd>

SMS:
cmd=timesync
cid=C_1234
address=011A0A30AAA0
address=011A0A30AAA1
```

La commande «scan» doit préciser les types d'informations demandées, dont:

- `data` : les données instantanées
- `rssi` : niveau RSSI
- `life-counter` : compteur de durée de vie de la batterie
- `rtc` : valeur de l'horloge RTC
- `tic` : profile de configuration

Pour «`rssi`», «`life-counter`» et «`rtc`», la passerelle va également récupérer les informations du/des répéteur(s) sur le chemin des modules.

Pour « `data` », une commande alarme avec *error="none"* sera envoyé pour indiquer au serveur qu'une commande a été exécutée.

Exemple : Demande de la valeur de l'horloge RTC et du compteur de durée de vie de la batterie de deux modules Wavenis :

```
XML:
<cmd cid="C_1235">
   <scan mode='rtc life-counter'>
      <address>011A0A30AAA0</address>
      <address>011A0A30AAA1</address>
   </scan>
</cmd>

SMS:
cmd=scan
cid=C_1235
mode=rtc,life-counter
address=011A0A30AAA0
address=011A0A30AAA1
```

Exemple: Demande de données instantanées de tous les modules Wavenis connus :

```
XML:
<cmd cid='C_1236'>
   <scan mode='data'/>
</cmd>

SMS:
cmd=scan
cid=C_1236
mode=data
```

4.2 Commandes spécifiques Wavenis

Les commandes Wavenis spécifiques peuvent être envoyées à un module Wavenis connue en utilisant la commande "`wavenis`".

La liste des sous-commandes supportées est:

Sous-commande	Description
moduflow-open	Demande d'ouverture de la vanne associée à un moduflow
moduflow-close	Demande de fermeture de la vanne associée à un moduflow
moduflow-state	Demande d'état d'ouverture/fermeture de la vanne associée à un moduflow
raw	Envoi d'une commande brutye Wavenis

Les trois commandes *moduflow* peuvent être envoyées à n'importe quel module Wavenis connu. La passerelle va utiliser les répéteurs configurés pour le module. La passerelle ne vérifiera pas que le module supporte la commande mais signalera toute erreur.

```
XML:
<cmd cid='C_1237'>
   <wavenis subcmd='moduflow-open'>
      <address>011A0A30AAA0</address>
   </wavenis>
</cmd>

SMS:
cmd=wavenis
cid=C_1237
subcmd=moduflow-open
address=011A0A30AAA0
```

Le résultat de la commande sera signalé sur le serveur comme une alarme `wavenis_cmd`.
Le code de résultat contenu dans l'alarme pour les commandes *moduflow-open* et *moduflow-close* peut être: `ok` (la commande a été acceptée par le module), `error` (la commande a été rejetée par le module) et `unsupported` (la commande n'est pas supportée).
Le code de résultat contenu dans l'alarme pour la commande *moduflow-state* peut être: `open` (la vanne est ouverte), `close` (la vanne est fermée) et `unsupported` (la commande n'est pas supportée).

La commande `raw` contient un paramètre additionnel qui doit être une chaine hexadécimale débutant par la commande applicative Wavenis. Elle sera envoyée dans une requête Wavenis REQ_SEND_FRAME.
Par exemple, pour lire l'octet de statut applicatif (0x20) d'un module connu (011A0A30AAA0), vous pouvez utiliser la commande applicative Wavenis 0x10 (Read parameter):

```
XML:
<cmd cid='C_1240'>
   <wavenis subcmd='raw' data='10012001'>
      <address>011A0A30AAA0</address>
   </wavenis>
</cmd>
SMS:
cmd=wavenis
cid=C_1240
subcmd=raw
data=10012001
address=011A0A30AAA0
```

Le code résultat contenue dans l'alarme de la commande *raw* peut être : `ok` (la commande a été acceptée par le module), `error` (la commande a été rejetée par le module). Si la commande a été acceptée par le module, la réponse est incluse dans l'alarme :

```
<wavenis_cmd_alarm>
      <date>2011-05-27T20:00:00</date>
      <cid>C_1240</cid>
      <source>ftp</source>
      <subcmd>raw</subcmd>
      <address>011A0A30AAA0</address>
      <result>ok</result>
      <request>10012001</request>
      <response>9001200106</response>
</wavenis_cmd_alarm>
```

4.3 Commandes spécifiques Modbus

Les commandes spécifiques Modbus peuvent être envoyées à un module Modbus module connu en utilisant la commande "*modbus*".
La liste des commandes supportées est :

Sous-commande	Description
write	Ecrire une valeur sur un équipement Modbus

Les adresses des variables doivent être formatées tel que expliqué au §11.4.

4.4 Commande Status

Exemple : Demande d'envoi du status de la passerelle :

```
XML:
<cmd cid='C_1237'>
   <status/>
</cmd>

SMS:
cmd=status
cid=C_1238
```

Les informations suivantes sont renvoyées au demandeur :

Nom XML	Nom SMS	Description
-	uid	Identifiant unique de la passerelle
/app/version	version	Version logicielle de la passerelle
/app/kernel	kernel	Linux kernel version.
/system/power	power	Présence de l'alimentation externe (booléen)
/system/defaults	defaults	Codes défauts séparés par des virgules
/com/modem/model	m_model	Nom du model de modem
/com/modem/firmware	m_version	Version du firmware du modem
/com/modem/imei	imei	International Mobile Equipment Identity
/com/modem/msisdn	msisdn	Mobile Subscriber ISDN Number (si disponible)
/com/modem/rssi	rssi	Puissance du signal reçu en dBm
/com/modem/csq	csq	Qualité de signal (CSQ, BER)
/com/modem/ip	m_ip	Adresse IP de la passerelle sur l'interface modem (ou dernière adresse attribuée).
/com/ethernet/ip	e_ip	Adresse IP de la passerelle sur l'interface Ethernet
/com/upload/last	u_last	Date de la dernière connexion au serveur distant réussie (périodique ou déclenchée)
/com/upload/next	u_next	Date de la prochaine connexion périodique au serveur distant
/wavenis/address	w_addr	Adresse Wavenis de la passerelle
/wavenis/last	w_last	Date de la dernière communication Wavenis réussie
/wavenis/modules/count	w_count	Nombre de modules Wavenis

Lorsque la commande d'état a été envoyée par SMS, le statut est renvoyé dans un SMS multiple avec une variable par ligne (nom = valeur).
Lorsque la commande d'état vient de INBOX (FTP ou WS), le fichier XML est téléchargé sous forme d'un fichier xml de supervision.

4.5 Commande de mise à jour

Commande de mise à jour du firmware de la passerelle :

```
XML:
<cmd cid='C_1238'>
   <update>
      <firmware>wrf_wavenis_v101.bin</firmware>
      <checksum>c1fb7d81f3d53a8b7bf94098115249d3</checksum>
   </update>
</cmd>

SMS:
cmd=update
cid=C_1237
firmware=wrf_wavenis_v101.bin
checksum=c1fb7d81f3d53a8b7bf94098115249d3
```

Le fichier du firmware doit être disponible dans le répertoire BIN sur le serveur FTP (voir paragraphe 3.2). Le checksum correspond au checksum md5 du fichier.

4.6 Commande d'un contact sec (sortie numérique)

Exemple: Ouvrir un contact sec de la passerelle (sortie numérique) :

```
XML:
<cmd cid='C_1239'>
   <d_output subcmd='open'/>
</cmd>
SMS:
cmd=d_output
cid=C_1239
subcmd=open
```

5 Configuration de la passerelle

5.1 Paramètres

Les paramètres de la passerelle sont traités de manière structurée. La configuration peut être exportée dans un fichier XML. L'installation d'une nouvelle configuration et la modification de la configuration actuelle sont réalisées en utilisant un fichier XML avec le même format.
Le schéma XML de configuration est présenté dans le paragraphe 18.

Un exemple de fichier XML de configuration est présenté dans le paragraphe 19.

Les principaux paramètres de la passerelle sont énumérés ci-dessous (en gras sont indiqués les valeurs par défaut configurées en usine) :

Nom	Valeur	Description
/uid	**6 derniers digits de l'adresse MAC Ethernet**	Identifiant unique de la passerelle
/name	**Adresse MAC préfixé de « WGRF_ »**	Nom de la passerelle (uniquement informatif)
/enable_local_config	false, **true**	Activation/désactivation de la configuration locale **non-implémenté**
/com/modem/pin/mode	**off**, manual, automatic	Mode de gestion du code PIN de la carte SIM (voir §3.1.1) **automatic non-implémenté**
/com/modem/pin/code	**0000**	Code PIN de la carte SIM
/com/modem/call_number	***99***1#**	Numéro d'appel de la connexion GPRS
/com/modem/apn		APN
/com/modem/login		Login APN
/com/modem/password		Mot de passe APN
/com/modem/mode	**ondemand,** alwayson, alwaysoff, off	Voir la description des modes §3.1). **alwayson non-implémenté**
/com/modem/delay	**60**	Délais en secondes avant déconnexion dans le mode de connexion *ondemand* **non-implémenté**
/com/modem/whitelist/caller_id		Liste blanche de numéros d'appelants autorisés pour la réception de SMS de commande. Si vide, pas de vérification
/com/ethernet/use_dhcp	**false**, true	Activation/désactivation du client DHCP
/com/ethernet/ip	**192.168.1.12**	Adresse IP de la WebdynRF
/com/ethernet/netmask	**255.255.255.0**	Masque de réseau IP
/com/ethernet/gateway		Adresse IP de la passerelle réseau
/com/ethernet/dns/server		Adresse IP du serveur DNS
/com/keepalive/method	icmp, tcp, **off**	Méthode Keepalive (***non-implémenté***)

/com/keepalive/address		***non-implémenté***
/com/keepalive/port	**5000**	***non-implémenté***
/com/keepalive/period	**600**	***non-implémenté***
/com/keepalive/timeout	**30**	***non-implémenté***
/com/request/upload	false, **true**	Connexion au serveur distant sur appui du bouton *Request*
/com/request/include_status	false, **true**	Téléchargement du status de la passerelle sur le serveur distant sur appui du bouton *Request*
/com/request/sms_status_recipient		Destinataire du SMS de status envoyé sur appui du bouton *Request*
/com/time/ntp/server		Adresse du serveur NTP
/com/time/timezone		Fuseau horaire local (utilise le nom zoneinfo standard tel que "Europe/Paris")
/com/time/alarm_threshold	**0**	Seuil d'alarme de désynchronisation en secondes (0=off).
/com/ftp/address		Adresse du serveur FTP
/com/ftp/login		Identifiant FTP
/com/ftp/password		Mot de passe FTP
/com/ftp/mode	**passive**, active	Mode FTP
/com/ftp/secured	**false**, true	Activation/Désactivation du protocole FTP sécurisé ***FTPS non implémenté***
/com/ftp/root_path	**/**	Répertoire racine sur le serveur FTP
/com/ftp/ws_notification	**none**, put, get, both	Mode notification WS de téléchargement FTP de fichier
/com/ws/address		Adresse WS ***ws non implémenté***
/com/ws/login		Identifiant WS ***ws non implémenté***
/com/ws/password		Mot de passe WS ***ws non implémenté***
/com/ws/secured	**false**, true	Activation/Désactivation SSL/TLS (HTTPS) pour WS. ***ws non implémenté***
/upload/config/method	**ftp**|ws|none	Méthode utilisée pour le téléchargement de la configuration ***ws non implémenté***
/upload/alarm/method	**ftp**|ws	Méthode utilisée pour le téléchargement des alarmes

		ws non implémenté
`/upload/supervision/method`	**ftp**|ws	Méthode utilisée pour le téléchargement des données de supervision ***ws non implémenté***
`/upload/data/method`	**ftp**|ws	Méthode utilisée pour le téléchargement des données
`/upload/data/format`	xml|**csv**	Format utilisé pour les données téléchargées
`/upload/data/schedule`		ID du schedule utilisé pour le téléchargement des données
`/alarm/*`		Configuration du moteur d'alarme (voir $13).
`/scheduler/*`		Configuration des schedules (voir $7).
`/wavenis/*`		Configuration Wavenis (voir $6.3).
`/metering/*`		Configuration Metering (voir $**Erreur ! Source du renvoi ntrouvable.**).
`/rfid/*`		Configuration Active RFID (voir paragraphe 0).
`/system/log/level`	7	Niveau de log (voir $14).
`/system/password/admin`	`high`	Mots de passe pour l'accès aux services HTTP et FTP locaux.
`/system/password/install`	`medium`	
`/system/password/data`	`low`	
`/system/ports/*`		Voir $0

La configuration peut être modifiée localement ou à distance. Toute modification déclenche le téléchargement de la nouvelle configuration sur le serveur.

Précisions sur les numéros de téléphone :

`/com/modem/whitelist`

- Si la liste est vide, tous le sn uméros sont considérés comme valides.
- `/com/modem/whitelist/caller_id`
- `/com/request/sms_status_recipient`
- Les numéros de téléphone doivent être écrit en format international.
- Ils doivent commencer par + et le code du pays.

5.2 Configuration par SMS

La configuration initiale de la passerelle peut être faite par SMS. Notamment, les paramètres de connexion peuvent être envoyés par SMS. Une fois cette configuration initiale terminée, le serveur distant peut compléter la configuration de la passerelle.

La première ligne du SMS doit contenir la commande «`CMD=config`».

Les lignes suivantes doivent avoir le format «`SHORTNAME=VALEUR`». Le SHORTNAME est constitué des premières lettres de chaque élément composant le nom du paramètre :
Par exemple, le SHORTNAME de «`com/modem/login`» est `LMC`.

Le contenu SMS est soumis aux règles suivantes:

- Les caractères espace en fin de ligne sont ignorés.
- Les SHORTNAME ne sont pas sensibles à la casse.
- Les valeurs booléennes, `true` et `false`, peuvent être remplacés respectivement par 0 et 1.
- Le retour chariot peut être remplacé par un point-virgule, mais les deux ne peuvent pas être mélangés dans un même SMS.
- Dans le cas de l'utilisation du caractère ';' comme séparateur de variables, il n'est pas possible d'utiliser ce caractère dans les valeurs des variables.

La longueur d'un SMS est limitée à 160 caractères.

Seuls les paramètres principaux et de communication peuvent être modifiés par SMS :

- /uid
- /name
- /enable_local_config
- /com/

Exemple :
Pour réaliser la configuration initiale d'une passerelle avec le contexte suivant:

- APN « m2minternet » ne nécessitant pas un identifiant / mot de passe.
- Communication avec le serveur distant de type FTP (168.112.23.123) en mode passif.

Vous pouvez envoyer le SMS suivant:

```
CMD=config
CMA=m2minternet
CFA=168.112.23.123
CFL=login
CFP=password
```

Remarque : Tous les paramètres utilisant leur valeur par défaut ont été omis.

Sur réception de ce SMS, la passerelle applique les paramètres et se connecte au serveur distant pour déposer le fichier de configuration qui en résulte. A partir de là, la passerelle peut être configurée à distance comme décrit ci-dessous.

Pour construire une liste (par exemple `/com/modem/whitelist/caller_id`), la variable concernée doit être répétée dans le SMS. Si la variable apparaît au moins une fois, la liste actuelle est remplacée. Si elle apparaît qu'une seule fois et sans valeur, la liste actuelle est effacée.

Exemples :

```
CMD=config
CTNS=1.2.3.4
```

Après le traitement de ce SMS, la passerelle va utiliser le serveur DNS 1.2.3.4.

```
CMD=config
CTNS=1.2.3.4
CTNS=1.2.3.5
```

Après le traitement de ce SMS, la passerelle va utiliser 2 serveurs DNS 1.2.3.4 et 1.2.3.5.

```
CMD=config
CTNS=
```

Après le traitement de ce SMS, la passerelle ne va plus utiliser aucun serveur DNS.

5.3 Configuration locale

La passerelle peut être configurée localement à travers une interface Web.

L'utilisation de l'interface Web est décrite dans le manuel d'installation de la passerelle WebdynRF (MI-WebdynRF.pdf).

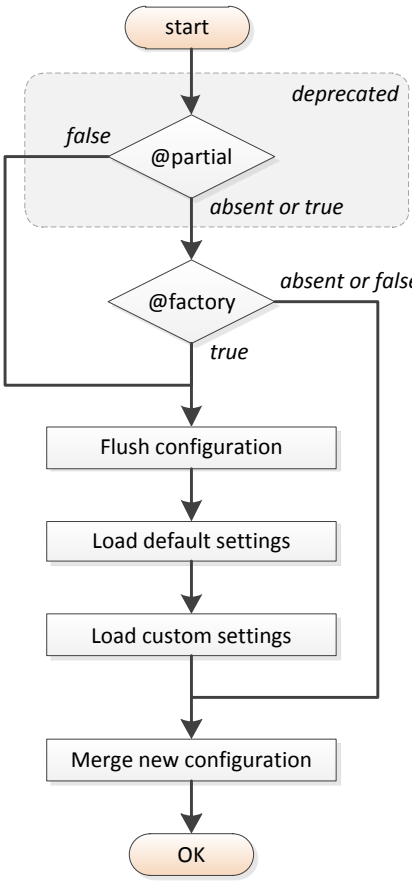
5.4 Configuration distante

Le serveur distant peut modifier la configuration en plaçant un fichier de configuration XML dans le répertoire INBOX sur le serveur FTP ou par signification à travers le service Web. Le format XML est utilisé dans les deux cas.

WS non implémenté

Le fichier XML est traité comme une nouvelle configuration si l'attribut XML `factory` est présent et égale la valeur `true`.

L'attribut `partial` est toujours supporté mais obsolète. L'attribut `factory` égal à `true` est équivalent à l'attribut `partial` égal à `false`.



Lorsque `factory` n'est pas présent ou égale à `false`, seuls les valeurs des paramètres de configuration présents dans le nouveau fichier de configuration sont mises à jour.
Lorsqu'une liste est présente dans le nouveau fichier de configuration, la liste entière est remplacée. C'est notamment le cas pour la liste des modules Wavenis et les schedules.
Par exemple, si `/config/Wavenis/modules` est présente dans le fichier de configuration, son contenu va remplacer la liste des modules précédemment configurée.

5.5 Contrôle de l'accès local

L'accès aux services HTTP et FTP locaux sont protégés par login/mot de passe. Toute tentative d'utilisation de ces services est enregistrée.
Trois niveaux d'accès sont définis : « `admin` », « `install` » et « `data` ».
L'administrateur (`admin`) possède les droits d'accès en lecture/écriture à tous les paramètres de configuration, en lecture pour les informations d'état de la passerelle et peut déclencher des actions.
L'installateur (`install`) possède les droits d'accès en lecture/écriture aux paramètres de configuration des modules finaux, notamment l'activation/désactivation du mode bridge Wavenis, en lecture pour les informations d'état de la passerelle et peut déclencher des actions.
L'utilisateur des données (`data`) possède seulement le droit d'accès en lecture aux informations d'état de la passerelle.

	Configuration passerelle	Etat passerelle	Configuration LAN	Actions
admin	R/W	R	R/W	Yes
install		R	R/W	Yes
data		R		No

Le mot de passe associé aux niveaux d'accès sont configurés dans `/system/password`. Ils peuvent être modifiés uniquement par un fichier de configuration depuis le serveur distant ou localement par l'administrateur.

Il est fortement recommandé de modifié ces mots de passe avant déploiement.

Restriction sur les mots de passe : ils ne doivent pas contenir les caractères ", &, ', <, >, ?, `.

Lorsqu'un fichier de configuration est téléchargé sur le FTP local par l'installateur (`install`), il sera rejeté s'il contient des paramètres n'appartenant pas à la configuration LAN.

Le port du bridge TCP Wavenis n'est pas protégé par mot de passe. En revanche, il doit être activé par l'installateur pour une période d'une heure maximum. Durant cette période, l'installateur peut initier des connexions au bridge. A l'expiration du délai, il ne sera plus possible d'initier de nouvelles connexions. Cependant, si une connexion est active, elle le restera.

5.6 Configuration des ports

Les paramètres suivants dans `/config/system/ports` sont utilisés pour configurer le mode de fonctionnement des ports :

Nom	Valeur	Description
rs232/mode	**off**, mbus	Mode RS232
rs485/mode	**off**	Mode RS485 (voir ci-dessous)
input_1/mode	**d_input**, pulse	Mode entrée numérique
input_2/mode	**d_input**, pulse	Mode entrée numérique
input_3/mode	**d_input**, pulse	Mode entrée numérique

Les paramètres du port RS485, dans */config/system/ports/rs485/*, sont :

Nom	Valeur
Mode	**off**, modbus
baudrate	4800, 9600, **19200**, 38400, 57600, 115200
Data	8
parity	odd, **even**, none
stop_bit	**1**, 2

Remarque : La spécification Modbus précise que si aucun bit de parité n'est utilisé, 2 bits de stop bits doivent être utilisés.

6 Wavenis

6.1 Mode transparent

La passerelle dispose d'un mode transparent TCP / Wavenis. Ce mode permet à l'utilisateur d'utiliser la passerelle comme un Waveport virtuel localement sur un PC connecté à la passerelle via Ethernet. Ce mode n'est pas destiné à être utilisé à distance.

Ce mode peut être désactivé par la valeur `false` du paramètre :
`/wavenis/bridge/enabled`

Lorsque ce mode est activé, dès que la connexion TCP est établie sur le port TCP dédié (`/wavenis/bridge/port`), la passerelle arrête d'utiliser la Wavecard à des fins de collecte de données et redirige toutes les communications entre la Wavecard et la connexion TCP.
Un logiciel pour Microsoft Windows est fourni (Toolbox). Ce logiciel va établir la connexion TCP avec la passerelle et va créer un port série virtuel. Il est alors possible d'utiliser la passerelle comme un Waveport virtuel avec les outils standards Wavenis.

Lorsque le mode transparent est en cours d'utilisation, toutes les données Wavenis reçues seront transmises au client TCP, y compris les alarmes émises à partir d'un module Wavenis.

L'utilisation du mode Wavenis transparent et de l'outil Toolbox est décrite dans le manuel d'installation de la passerelle WebdynRF (MI-WebdynRF.pdf).

6.2 Modules Wavenis supportés

La passerelle supporte les modules Wavenis suivants :

- Waveflow (1, 2 et 4 entrées)
- Wavetherm Dallas (1 et 2 entrées), PT100 (1 entrée) and PT1000 (1 entrée)
- Wavesense 4-20mA (1 entrée) and 0-5V (1 entrée)
- Wavelog (2 et 4 entrées)
- Wavetic (1 et 2 entrées)

La passerelle ne gère pas la configuration de ces modules.

6.3 Configuration Wavenis

Nom	Valeur	Description
/wavenis/bridge/enabled	**false**, true	Activation/désactivation du mode transparent Wavenis
/wavenis/bridge/port	**4000**	Port TCP utilisé par le mode transparent Wavenis
/wavenis/time/mode	utc,local,**nodst**	Mode de gestion du temps (voir §6.5)
/wavenis/alarm/mode	**basic,** extended	Mode de gestion des alarmes Wavenis ***Extended* non-implémenté**
/wavenis/alarm/sources/unknown	on, **off**, delayed	Activation/désactivation des alarmes émises lorsque des données sont reçues d'un module inconnu
/wavenis/alarm/sources/route	**on**, off, delayed	Activation/désactivation des alarmes émises lorsque des données sont reçues d'un module connu mais ne suivant pas le chemin correct
/wavenis/modules/*		Liste ordonnée des modules Wavenis

Chaque module est configuré comme suit :

Nom	Description
module/address	Adresse Wavenis
module/label	Nom du module (uniquement informatif)
module/type	Type de Module
module/repeaters	Liste de répéteurs
module/mode	Mode Data (immediat, datalog).
module/nbinput	Nombre d'entrées du module.
module/schedule	ID du schedule utilisé pour l'acquisition de données

Toute adresse Wavenis peut être donnée dans sa forme hexadécimale (12 chiffres) ou dans sa forme décimale (15 chiffres, avec un tiret optionnel après les 5ème et 7ème chiffres).
Les répéteurs doivent être donnés de la passerelle vers le module.
Les données sont demandées à tous les modules associés à un schedule donné dans l'ordre configuré. Une requête pour un module donné est répétée jusqu'à trois fois si nécessaire.

6.4 Alarmes Wavenis

Lorsque la passerelle reçoit un message d'alarme Wavenis, il l'acquitte (au module expéditeur).
Si l'alarme se déclenche à partir d'un module connu, la passerelle la traite et initie ensuite une connexion vers le serveur distant pour la déposer.
Si l'alarme provient d'un module inconnu, une alarme *Wavenis_unknown* est déclenchée. Cette alarme sera envoyée qu'une seule fois par module et peut être immédiatement envoyée (*on*) ou à la prochaine connexion (*delayed*).
Si l'alarme se déclenche à partir d'un module connu mais a suivi un chemin radio différent de celui configuré, une alarme *Incorrect route* est déclenchée.
Cette alarme sera envoyée qu'une seule fois par module et peut être immédiatement envoyée (*on*) ou à la prochaine connexion (*delayed*).
L'alarme Wavenis originale sera toujours acquittée. La passerelle ne filtre pas les alarmes Wavenis (càd il n'y a pas de paramètres permettant de sélectionner les alarmes d'un module donné). Elles peuvent être néanmoins activés et / ou désactivé lors de la configuration du module lui-même.

Le mode de traitement des alarmes disponible est le mode basique. Dans ce mode, les informations envoyées au serveur sont uniquement celles contenues dans la trame d'alarme.

Le mode étendu n'est pas implémenté.

L'alarme est téléchargée vers le serveur distant en XML comme spécifié par le schéma d'alarme XML (voir paragraphe 20).

6.5 La gestion du temps Wavenis

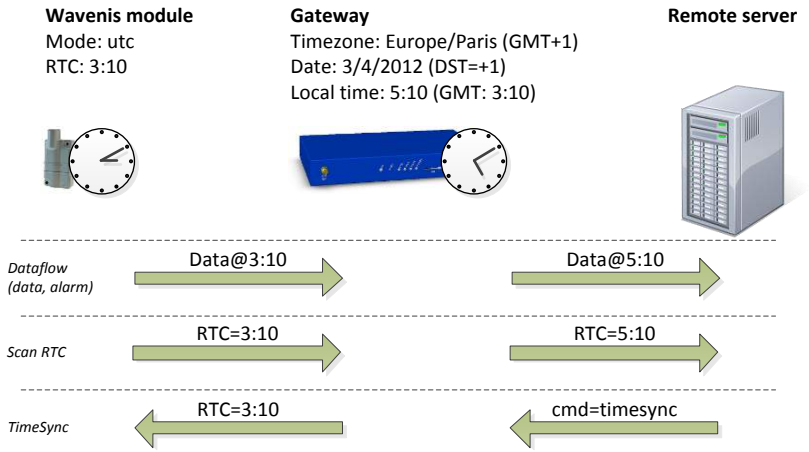
La passerelle est configurée avec un fuseau horaire donné, qui peut inclure l'heure d'été (DST : Daylight Saving Time) (typiquement 1 heure supplémentaire de décalage GMT / UTC pendant l'été). Toutes les données manipulées par la passerelle sont horodatées avec l'heure locale.
L'horodatage des données des modules Wavenis utilisent leur propre horloge RTC.
Puisque les modules Wavenis ne gèrent pas l'heure d'été, il est nécessaire d'indiquer à la passerelle comment manipuler leurs horloges RTC.

Le paramètre `/wavenis/time/mode` peut prendre les valeurs suivantes :

- `utc` : L'horloge RTC des modules Wavenis sont mis à l'heure UTC/GMT (Coordinated Universal Time, Greenwich Mean Time).
- `local` : L'horloge RTC des modules Wavenis sont fixés sur le même fuseau horaire que la passerelle en tenant compte de l'heure d'été.
- `nodst` : L'horloge RTC des modules Wavenis sont fixés sur le même fuseau horaire que la passerelle sans tenir compte de l'heure d'été. Il s'agit de la valeur par défaut.

Le mode `local` implique que les horloges RTC des modules Wavenis doivent être réglés après chaque changement d'heure d'été (généralement deux fois par an). Ceci peut être réalisé avec la commande timesync. C'est le mode à utiliser si le schedule des modules Wavenis doit être exécuté en fonction de l'heure locale. Notez que dans ce mode, il y aura une erreur de décalage dans les données générées entre un changement d'heure d'été et l'appel associé à timesync.
Dans tous les cas, la passerelle convertit tous les horodatages contenus dans les données Wavenis à son heure locale (les horodatages des journaux, les horodatages d'alarme, les lectures rtc). La passerelle prend également en compte le mode de temps lors du réglage de l'horloge RTC d'un module Wavenis.

Pour illustrer cela, prenons un cas où le mode est réglé sur `utc` :



6.6 Support des modules WaveTIC

La configuration d'un module WaveTIC nécessite les mêmes paramètres que les autres modules Wavenis. Néanmoins, contrairement à ces derniers, la passerelle se base sur des paramètres additionnels obtenus directement du module.

De plus, contrairement aux autres types, en mode datalog, seules les nouvelles entrées seront récupérées à partir du module. La passerelle assure le suivi de la dernière récupérée du module.
Lorsqu'un nouveau module est configuré, la passerelle interroge le module pour obtenir cette information. Ceci est accompli par le déclenchement d'une commande *"scan mode=tic"* (voir §4.1). Le résultat de cette commande est envoyé au serveur distant.
La commande ""scan mode=tic"" permet de récupérer les informations suivantes:

- Le profil TIC sélectionné
- Les détails du son profil personnalisé sélectionné
- La valeur des étiquettes TIC statiques comme ADCO (numéro de série du compteur)

Cette commande réinitialise également le point de départ du datalog.

Si le profile TIC d'un module configure est modifié par la suite, il est nécessaire d'utiliser la commande "scan mode=tic". Jusqu'à la fin de cette commande, la passerelle va détecter l'incohérence et retourner une erreur «err_config" (voir paragraphe suivant).

6.7 Erreurs de communication Wavenis

Pendant le processus de collecte des données, si un module ne répond pas, le fichier de données XML contiendra une entrée avec un élément «*err_status*" indiquant l'origine de l'erreur qui peut être :

Nom	Description
none	Pas d'erreur (omis)
no_response	Le module n'a pas répondu
err_repeater_ 1	Le premier répéteur n'a pas répondu
err_repeater_ 2	Le deuxième répéteur n'a pas répondu
err_repeater_ 3	Le troisième répéteur n'a pas répondu
err_config	La réponse n'est pas cohérente (WaveTIC)

De plus, l'entrée contient un élément "*retry_count*» contenant le nombre de nouvelle tentative si au moins une tentative était nécessaire.

7 Impulsions

Les entrées numériques peuvent être sélectivement configurées en compteurs d'impulsions (voir §0). En mode compteur d'impulsions, un compteur associé va s'incrémenter après chaque impulsion de plus de 10ms. La valeur courante sera sauvegardée pour chaque occurrence du schedule spécifié. Les paramètres suivants sont configurés dans `/config/metering/pulse` :

Nom	Description
Schedule	ID du schedule pour l'acquisition des impulsions
input_1/label	Nom de l'entrée (uniquement informatif)
input_1/unit	Unité (et poids) d'impulsion (uniquement informatif)
input_2/label	Nom de l'entrée (uniquement informatif)
input_2/unit	Unité (et poids) d'impulsion (uniquement informatif)
input_3/label	Nom de l'entrée (uniquement informatif)
input_3/unit	Unité (et poids) d'impulsion (uniquement informatif)

Les paramètres « `label` » et « `unit` » sont ajoutés dans les données enregistrées avec la valeur d'index.

8 Wired M-Bus

Des données peuvent être collectées d'un équipement M-Bus. Pour cela, un transceiver M-Bus doit être connecté au port RS232 et ce port doit être configuré en mode M-Bus (voir §0).
Les équipements M-Bus doivent être configurés avec une adresse unique sur le bus.
Un scan du bus doit être initié depuis l'interface web (voir manuel d'installation). Les équipements M-Bus découverts durant ce scan seront interrogés à chaque occurrence du scheduler associé. Si des équipements sont enlevés ou ajoutés sur le bus, un nouveau scan doit être initié afin que la passerelle prenne en compte la modification.

Les paramètres suivants sont configurés dans `/config/metering/mbus` :

Name	Description
schedule	Schedule ID for M-Bus data collection

9 Wireless M-Bus

En version de carte radio Wireless M-Bus la passerelle WebdynRF peut recevoir des données provenant de modules Wireless M-Bus connus en mode S1 ou T1.

Les paramètres suivants sont configurés dans `/config/metering/wmbus` :

Nom	Valeur	Description
mode	S1, **T1**	Mode Wireless M-Bus
long_preamble	true, **false**	Longueur du préambule Radio (ignoré en mode T)
modules/*		Liste des modules Wireless M-Bus

Si le cryptage OMS est activé, le nombre de modules est limité à 64. Les modules sans les clés de chiffrement seront ignorés.
Chaque module est configuré comme suit :

Nom	Description
module/address	Adresse Wireless M-Bus
module/label	Nom du module (uniquement informatif)
module/key	Clé de cryptage du module

10 Tags RFID actifs

En version de carte radio RFID, la passerelle peut:

- Surveiller la présence de tag RFID actifs dans sa zone de couverture.
- Recueillir des données provenant de tag RFID actifs.

Le récepteur est compatible avec les tags actifs de *ELA Innovation*.
http://www.ela.fr

Les tags doivent être configurés en mode 24 bits avec checksum radio de 16 bits.
La passerelle reçoit la transmission périodique de l'ensemble des tags RFID actifs.
Un décalage de CRC optionnel peut être configuré pour filtrer tous les tags qui ne sont pas configurés avec ce même décalage.
Tous les ID reçues avec un RSSI au-dessus du seuil configuré sont ignorés. Ce seuil permet de réduire la zone de couverture.

Remarque : La gamme RSSI est de 110 à 200. Si le seuil est fixé à une valeur inférieure à 110, aucune donnée ne sera reçue. Si le seuil est fixé à une valeur supérieure à 200, toutes les données reçues sont traitées.

Si le tag est de type ID (le bit de poids fort de l'ID est égal à 0), les 3 bits suivants sont traités comme des indicateurs d'alarme et ne sont donc pas considérés comme faisant partie de l'ID.
Si le tag est de type ID+DATA (le bit de poids fort de l'ID est égal à 1), les 12 premiers bits sont utilisés comme l'ID et les 12 bits suivants sont considérés comme des données.

La passerelle gère une liste de tags dans sa zone de couverture. Un tag entrera dans cette liste seulement si son ID a été reçue régulièrement pendant le délai de détection «*entering*». Une étiquette sera retirée de cette liste dès que son ID n'a pas été reçue pendant le délai de détection de «*leaving*».
Lorsqu'un tag ID+DATA est considéré comme étant dans la zone de couverture, ses données seront enregistrées et téléchargées sur le serveur distant.
Les alarmes peuvent être envoyées immédiatement ou lors de la prochaine connexion pour les événements suivants : entrer d'un tag, sortie d'un tag, un indicateur est défini dans l'ID d'un tag de type ID.

Chaque fois que la passerelle se connecte au serveur distant, elle envoie :

- Toutes les données recueillies à partir de tags ID+DATA.
- Une liste de tous les tags dans la zone de couverture.
- Toutes les alarmes différées.

Les paramètres suivants sont disponibles dans */config /rfid* :

Nom	Valeur	Description
rssi_threshold	0-**255**	Niveau RSSI de filtrage des tags
Crc	**0**	Décalage de CRC optionnel
detection_delay/entering	**600**	Délais avant qu'un tag soit considéré dans la zone
detection_delay/leaving	**500**	Délais avant qu'un tag soit considéré hors zone
alarm/sources/entering	**on**, off, delayed	Alarme lorsqu'un tag entre dans la zone
alarm/sources/leaving	**on**, off, delayed	Alarme lorsqu'un tag sort de la zone
alarm/sources/id_flags	on, off, **delayed**	Alarme lorsqu'un tag ID présente un indicateur est défini

Remarque : Les données reçues de tags ID+DATA sont enregistrés sous forme de données brutes. Elles ne sont pas converties en température / humidité / mouvement car la passerelle n'est pas au courant de ce type d'information. En outre, les valeurs spécifiques de type "*end of battery life*" ne sont pas reconnues (puisque cette valeur dépend de son type, 0x7FF pour les tags T/HR et 0xFFF pour les tags MOV).

11 Modbus

La passerelle peut fonctionner comme un dispositif maître Modbus. Cette fonctionnalité permet de lire / écrire dans les registres sur des modules Modbus RTU et TCP esclaves.
La configuration Modbus de la passerelle consiste en une liste d'ensembles de données et une liste de modules. Un ensemble de données est une liste de registres pour un type donné

d'équipement esclave Modbus. La liste des modules associe un module esclave Modbus à un ensemble de données et un scheduler.
En mode polling, la valeur de toutes les variables sont continuellement mises à jour. Ces valeurs peuvent être surveillées pour détecter des changements ou les comparer à des seuils.

Les valeurs courantes seront enregistrées lorsque :

- La valeur d'une donnée surveillée change ou franchit un certain seuil,
- Le schedule associé se produit.

En mode d'instantané (i.e. pas en mode polling), les valeurs de toutes les variables sont mises à jour et enregistrées lorsque et seulement lorsque le schedule associé se produit.
Indépendamment du processus de collecte des données Modbus, il est possible d'écrire dans les registres de certains modules esclaves à l'aide de la commande Modbus (voir §0).

11.1 Configuration

La configuration Modbus dans */config/modbus* contient les paramètres suivants :

Nom	Valeur	Description
tcp/timeout	1000	Timeout de réponse Modbus/TCP en ms
rtu/timeout	1000	Timeout de réponse Modbus/RTU en ms
rtu/turnaround	100	Délais de turnaround Modbus/RTU en ms
datasets/*		Liste d'ensembles de données
modules/*		List de modules

En complément de ces paramètres, les paramètres */config/system/ports/rs485/* doivent être correctement configurés (voir §0). En particulier, le paramètre */config/system/ports/rs485/mode* doit être configuré égale à "modbus".

11.2 Ensembles de données Modbus

La configuration d'un ensemble de données (*/config/modbus/datasets/dataset*) consiste à configurer les paramètres suivants :

Nom	Description
id	Identifiant unique de l'ensemble de données Modbus
label	Nom de l'ensemble de données (uniquement informatif)
vars/*	Liste de variables
boundaries/*	Liste des registres limites
polling	Polling continue (*true* or *false*)

11.2.1 Variables

Chaque variable est définie dans */config/modbus/datasets/dataset/vars/var* par les paramètres suivants :

Nom	Description
name	Nom de la variable (uniquement informatif)
type	Type de variable (S0, S1, S3, S4)
address	Adresse de registre étendue 16-bit
size	Taille en bits pour *discrete input* et *coil*, en octets pour les registres
format	Voir liste ci-dessous
flags	Liste de flags séparés par des virgules (voir la définition du flag ci-dessous)
threshold/low	Niveau de seuil bas (voir ci-dessous)
threshold/high	Niveau de seuil haut (voir ci-dessous)
threshold/hysteresis	Hystérésis appliquée aux deux seuils

Paramètre "type"

Le type d'une variable est l'un des quatre types de registres Modbus.

Type	Description	Read (multiple)	Write (single)	Write (multiple)
S1	Discrete input	0x02	-	-
S0	Coil	0x01	0x05	0x0F
S3	Input register	0x04	-	-
S4	Holding register	0x03	0x06	0x10

Dans le tableau ci-dessous, les codes fonction de lecture / écriture sont donnés à titre indicatif. Les requêtes Modbus ne font pas partie de la configuration, mais seront déduites. En particulier plusieurs codes de fonction *Read* seront utilisées partout où cela réduit les coûts de communication.

Paramètre "address"

Ce document se réfère toujours aux adresses Modbus de registre (commençant à 0) et jamais au numéro de registre Modbus (commençant à 1).

Paramètre "format"

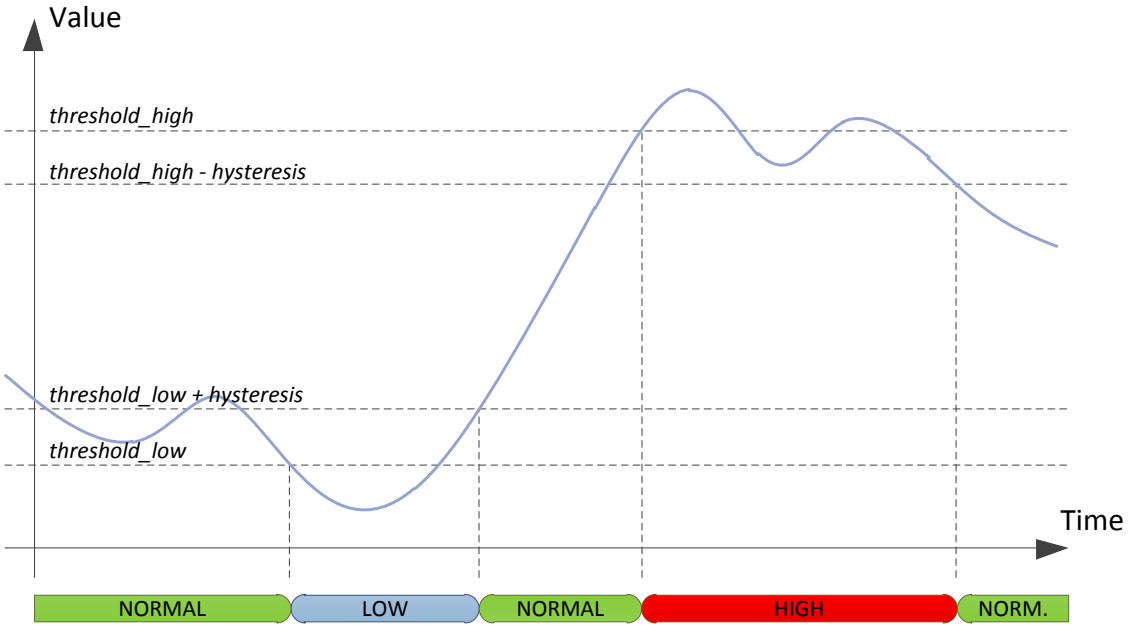
Format	Description	Coil	Register
raw	Les données seront représentées sous forme d'une chaine binaire pour les *discrete inputs* et les *coils* et d'une chaine hexadécimale pour les *registers*	✓	✓
boolean	Vrai ou faux	✓	
integer	16 ou 32-bit entier non-signé		✓
float	16 ou 32-bit à virgule flottante (IEEE 754)		✓
ascii	Chaîne de caractères ASCII		✓

Paramètre "flag"

Format	Description
cmd_only	La variable ne sera pas lue à partir du module Modbus, mais peut être écrite
little_endian	Interprète les deux registres de 16 bits d'une valeur de 32 bits en little-endian
no_opt	Une requête Modbus dédié sera utilisée pour lire cette variable.
is_status	En mode polling, tout changement de cette variable va déclencher une lecture de l'ensemble de données.
is_alarm	Comme *is_status* mais aussi déclenche une demande de connexion.

Paramètre "alarm"

Pour les variables flottantes et entières, deux seuils peuvent être définis (*alarm/low* et *alarm/high*) avec une valeur d'hystérésis. Chaque fois que la variable est mise à jour, sa valeur est comparée à ce niveau afin de déterminer un statut associé (*low*, *normal*, *high*), comme indiqué ci-dessous:



Lorsqu'au moins un niveau de seuil est défini, les flags *is_status* et *is_alarm* s'appliquent à l'état qui en résulte. Par exemple, si *is_alarm* est définie à *true*, la demande de connexion sera déclenchée seulement lorsque l'état de la variable change et pas à chaque fois que sa valeur change.

Données supplémentaires dans le mode polling :

En mode polling, des données supplémentaires seront maintenues pour les valeurs entières et flottante : Les valeurs min / max / moyennes et le nombre d'échantillons depuis le dernier enregistrement de données.

11.2.2 Boundaries

Non implémenté en V2.x.

11.3 Esclaves Modbus

Un module est une instance d'un ensemble de données pour une adresse Modbus donnée. La configuration d'un module Modbus (*/config/modbus/modules/module*) comprend les paramètres suivants:

Nom	Description
label	Nom uniquement informatif
dataset	Identifiant de l'ensemble de données
address	Adresse Modbus (1-247)
ip	Adresse IP (vide pour les équipements RTU)
schedule	Identifiant du schedule

Remarque : Les équipements Modbus/TCP doivent être configurés pour écouter le port Modbus TCP par défaut (502).

11.4 Adresse des variables

La commande Modbus utilise des adresses formatées comme expliqué ci-dessous :

Modbus/RTU

- <modbus_address>/<register_type>@<register_address>
- Exemple :
  Registre d'entrée à l'adresse 0x0056 sur l'équipement Modbus 45.
  => 45/S3@0x0056

Modbus/TCP

- <device_ip>:<modbus_address>/<register_type>@<register_address>
- Exemple:
  Registre d'entrée à l'adresse 0x0F0C sur l'équipement Modbus 223 on à l'adresse IP 192.168.0.17.
  => 192.168.0.17:223/S3@0x0F56

Remarque : La *modbus_address* et le *register_address* peuvent être soit sous forme décimale ou hexadécimale. Ce dernier doit être précédé de "0x".

12 Scheduler

Le scheduler est en charge de toutes les tâches périodiques.
La configuration du scheduler consiste en une liste de schedules.
Chacun de ces schedules possède un identifiant (entier positive) unique qui est utilisé pour lier une tâche à un schedule spécifique.

Les schedules peuvent être utilisés indépendamment pour déclencher la collecte de données et télécharger des données recueillies.

Name	Description
/scheduler/schedules/schedule	La configuration de chaque schedule (voir ci-dessous) sera enregistrée sous cet élément.

Chaque schedule est configuré comme suit :

Nom	Description
schedule/id	Identifiant unique de schedule (Entier)
schedule/label	Nom uniquement informatif du schedule
schedule/type	Daily, Weekly, Monthly, Yearly ou Follower : voir description ci-dessous
schedule/parent	Référence au schedule parent pour un schedule de type Follower.
schedule/start/time	Heure de la première occurrence (non utilisé pour les schedules de type Yearly)
schedule/start/datetime	Date et heure de la première occurrence dans une période donnée (utilisé uniquement pour les schedules de type Yearly).
schedule/start/dayofweek	Numéro du jour dans la semaine de la première occurrence (1=Lundi, 7=Dimanche) (utilisé uniquement pour les schedules de type Weekly).
schedule/start/dayofmonth	Numéro du jour dans le mois de la première occurrence (utilisé uniquement pour les schedules de type Monthly).
schedule/interval	Interval entre les occurrences (en secondes)
schedule/count	Nombre d'occurences

Configuration des différents types de schedules :

Schedule de type Daily :
Chaque jour, la première occurrence T0 est donnée par l'heure renseignée dans *time.*

Le format de la variable time est le suivant : HH:MM:SS
Par exemple 09:30:00

Les occurrences suivantes interviendront à l'heure T_i :

$$T_i = T_0 + i \times \Delta t \qquad \begin{cases} i < count \\ \forall i \; jour(T_i) = jour(T_0) \end{cases}$$

Δt correspond à la valeur en secondes renseignée dans *interval*.

Schedule de type Weekly :
Chaque semaine, la première occurrence T0 est donnée par le jour de la semaine renseignée dans *dayofweek* et l'heure renseignée dans *time*.

Le format de la variable time est le suivant : HH:MM:SS
Par exemple 09:30:00

La variable *dayofweek* est un entier entre 1 et 7 (1=Lundi et 7=Dimanche)

Les occurrences suivantes interviendront à l'heure T_i :

$$T_i = T_0 + i \times \Delta t \qquad \begin{cases} i < count \\ \forall i \; semaine(T_i) = semaine(T_0) \end{cases}$$

Δt correspond à la valeur en secondes renseignée dans *interval*.

Schedule de type Monthly :
Chaque mois, la première occurrence T0 est donnée par le numéro de jour du mois renseigné dans *dayofmonth* et l'heure renseignée dans *time*.

Le format de la variable time est le suivant : HH:MM:SS
Par exemple 09:30:00

Les occurrences suivantes interviendront à l'heure T_i :

$$T_i = T_0 + i \times \Delta t \qquad \begin{cases} i < count \\ \forall i \; mois(T_i) = mois(T_0) \end{cases}$$

Δt correspond à la valeur en secondes renseignée dans *interval*.

Schedule de type Yearly :
Chaque année, la première occurrence T0 est donnée par la date renseignée dans *datetime.*

Le format de la variable *datetime* est le suivant : AAAA-MM-JJTHH:MM:SS
Par exemple, pour une première occurrence le 11 février 2012 à 13H00 :
datetime = 2012-02-11T13:00:00.

Les occurrences suivantes interviendront à l'heure T_i :

$$T_i = T_0 + i \times \Delta t \qquad \begin{cases} i < count \\ \forall i \ année(T_i) = année(T_0) \end{cases}$$

Δt correspond à la valeur en secondes renseignée dans *interval*.

Schedule de type Follower :
Un schedule de type «Follower» sera produit après la fin de chaque occurrence du schedule de référence. Le schedule *parent* ne peut pas être de type "Follower".
Ce type permet de déclencher par exemple un téléchargement des données après l'achèvement d'une collecte de données prévue.

Exemple :
Vous souhaitez collecter les données de tous les modules Wavenis une fois par jour à minuit et télécharger les données, juste après. Vous pouvez configurer un schedule de type « *Daily* » pour la collecte de données et un schedule de type « *Follower* » du premier schedule pour le téléchargement des données.

```
<schedules>
   <schedule>
      <id>1</id>
      <label>Data collect</label>
      <type>day</type>
      <start>
         <time>00:00:00</time>
      </start>
   </schedule>
   <schedule>
      <id>2</id>
      <label>Data upload</label>
      <type>follow</type>
      <parent>1</parent>
   </schedule>
</schedules>
```

Exemples :

Besoin	*type*	*time*	*Day of week*	*Day of month*	*Date time*	*interval*	*count*
Tous les mardi à 15:00:00	week	15:00:00	*2*			0	1
Tous les 2ème jour du mois à 00:00:00	month	00:00:00		*2*		0	1
Tous les jours à 14:00:00	day	14:00:00				0	1
Toutes les heures entre 8H00 et 18H00 tous les mardis	week	08:00:00	*2*			3600	11
Toutes les 2 heures entre 8H00 et 20H00 le 31 décembre	year				2012-12-31T08:00:00	7200	7

13 Moteur d'alarme

Le moteur d'alarme génère des alarmes basées sur des événements internes.
Chaque source d'alarme peut être activée individuellement et sera téléchargée immédiatement au serveur distant (`on`) ou à la connexion suivante (`delayed`).

Nom	Valeurs	Description
sources/power	on, **off**, delayed	Changement d'état de l'alimentation principale
sources/modem_ip	on, **off**, delayed	Changement d'adresse IP
sources/msisdn	on, **off**, delayed	Changement MSISDN
sources/sw_version	**on**, off, delayed	Changement de la version logicielle
sources/defaults/ignored	***<empty>***	Liste des codes défauts ignorés par la passerelle (séparés par des virgules)
sources/defaults/delayed	***<empty>***	Liste des codes défauts transférés à la connexion suivante (séparés par des virgules)
sources/d_inputs/*		Changement d'état des entrées TOR (voir ci-dessous).
sources/d_output		Changement d'état des sorties TOR (voir ci-dessous).

La passerelle génère également des alarmes de type défaut, dont les codes sont reportés ci-dessous :

Code	Description
D_MODEM	Défaut du modem
D_MODEM_PUK	Carte SIM bloquée
D_ETHERNET	Défaut de l'interface Ethernet
D_WAVENIS	Défaut radio Wavenis
D_RFID	Défaut du récepteur RFID
D_INTERNAL_BAT	Défaut batterie interne

Lorsque l'alarme "`power`" est activée, le moteur d'alarme enverra une alarme sur la perte et la récupération de son alimentation.

Lorsque l'alarme "`modem_ip`" est activée, le moteur d'alarme enverra une alarme contenant l'adresse IP de la passerelle chaque fois qu'elle change.

Lorsque l'alarme "`msisdn`" est activée, le moteur d'alarme enverra une alarme sur changement de la carte SIM.

Lorsque l'alarme "`sw_version`" est activée, le moteur d'alarme enverra une alarme contenant la version logicielle de la passerelle, après une mise à jour.

Des alarmes multiples peuvent être configurées pour les entrées numériques.
Une alarme pour une entrée numérique peut être configurée comme suit:

Nom	Valeur	Description
d_input/index		Index de l'entrée numérique
d_input/label		Nom de l'alarme (uniquement informatif)
d_input/mode	on, **off**, delayed	On : Envoi immédiat Off : Envoi désactivé Delayed : Envoi à la prochaine connexion
d_input/type	**none**, raising, falling, both	None : Détection désactivée Raising : Détection sur front montant Falling : Détection sur front descendant Both : Raising+Falling

Une alarme peut être configure pour la sortie numérique comme suit :

Nom	Valeur	Description
d_output/label		Nom de l'alarme (uniquement informatif)
d_output/mode	on, off, delayed	On : Envoi immédiat Off : Envoi désactivé Delayed : Envoi à la prochaine connexion
d_output/type	none, raising, falling, both	None : Détection désactivée Raising : Détection sur front montant Falling : Détection sur front descendant Both : Raising+Falling

Le front montant correspond à la fermeture du contact sec de la passerelle.

L'alarme est téléchargée vers le serveur distant en XML comme spécifié par le schéma d'alarme XML (voir paragraphe 20).

14 Fichiers de log

La passerelle enregistre les principaux événements dans des fichiers de log. La taille du fichier de log est limitée à 200Ko.
Sur demande, les fichiers de log précédents et actuels sont concaténés et téléchargés vers le serveur distant en tant que données de supervision.
Le fichier de log est un fichier texte avec une entrée log par ligne.

Chaque ligne est formatée comme suit:

[TIMESTAMP][LEVEL][SOURCE] EVENT

L'horodatage est un timestamp Unix (temps écoulé depuis l'EPOC) en secondes suivi par un point et les microsecondes.

Les événements suivants sont enregistrés:

Evènements	Formats
Schedules	Schedule X occurred
Défauts	Default X detected
	Default X cleared
PPP	PPP connecting
	PPP connected IP=X
	PPP connection failure
	PPP disconnect
FTP	FTP connecting
	FTP connected
	FTP connection failure
	FTP get X
	FTP put X
	FTP disconnect
SMS	SMS received from X
Commandes	Processing command X
Horloge	Clock synchronization delta=X
	NTP connection failure
Internes	Reboot
	Reboot modem

La liste n'est pas exhaustive et chaque ligne peut contenir des informations supplémentaires après celle spécifiée ci-dessus. Par exemple, lorsque survient un schedule, le fichier de log peut contenir «*Schedule X occurred, next is Y at Z*».

Le niveau de détail des logs peut être configuré par le paramètre /system/log/level. La valeur doit être comprise entre 8 (pas de log) à 1 (plus détaillé):

Level	Name	Description
8	None	Rien n'est loggé
7	Critical	Info critiques seulement
6	Error	
5	App	Niveau par défaut
4	Warning	
3	Notice	
2	Info	
1	Debug	Logs le plus détaillé

Le paramètre de niveau de log peut également être défini par source. Le format est alors :

```
default_level,source:level,source:level,…
```

Par exemple, pour utiliser le niveau 5 pour toutes les sources sauf pour la source Coronis avec le niveau 1 :

```
/system/log/level="2,Coronis:1"
```

15 Synchronisation de l'horloge interne

La passerelle synchronise l'horloge système en utilisant le protocole NTP.

Elle conserve l'heure en UTC (Temps Universel, aussi connu comme GMT) et calcule un temps local basé sur le fuseau horaire configuré. Elle gère l'heure d'été (DST).

Lorsque la connexion utilise le modem, la synchronisation se fait au début de chaque connexion au serveur, mais pas plus d'une fois par jour.

Une alarme est déclenchée lorsque la différence entre les deux horloges est plus grande qu'une valeur configurable.

Lorsque la connexion utilise Ethernet, un client NTP est activé sur la passerelle. Ce client va régler la vitesse de l'horloge système, afin de la maintenir synchronisée avec l'horloge du serveur NTP.

16 Mise à jour du firmware de la passerelle

Une mise à jour peut s'effectuer à distance. Le nouveau firmware doit être mis à disposition sur le serveur FTP dans le répertoire BIN dédié (voir paragraphe 3.2). Une commande mise à jour doit ensuite être envoyée à la passerelle.

Exemple : Commande de mise à jour du firmware de la passerelle avec le firmware `wrf_wavenis_v101.bin` présent dans le répertoire `BIN`.

```
XML:
<cmd cid='C_1238'>
   <update>
      <firmware>wrf_wavenis_v101.bin</firmware>
      <checksum>c1fb7d81f3d53a8b7bf94098115249d3</checksum>
   </update>
</cmd>

SMS:
cmd=update
cid=C_1237
firmware=wrf_wavenis_v101.bin
checksum=c1fb7d81f3d53a8b7bf94098115249d3
```

Le checksum correspond au checksum md5 du fichier.

17 Support

En cas de problèmes techniques relatifs à nos produits, contactez le support de WEBDYN :

Webdyn SA
26 Rue des Gaudines - 78100 Saint-Germain-en-Laye
Tel.: +33 1 39 04 29 40
Fax.: +33 1 39 04 29 41
Mail : support@webdyn.com
http://www.webdyn.com

Nous aurons besoin des éléments suivants :

- Numéro de série de la passerelle.
- Version de matériel et de logiciel de la passerelle.

18 ANNEXE A Schéma XSD – Configuration

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:wd="http://www.webdyn.com/WRF_config_20110606"
  targetNamespace="http://www.webdyn.com/WRF_config_20110606"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

<xsd:element name="config">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="uid" type="xsd:string" minOccurs="0"/>
      <xsd:element name="name" type="xsd:string" minOccurs="0"/>
      <xsd:element name="enable_local_config" type="xsd:boolean"
minOccurs="0"/>
      <xsd:element name="com" type="wd:com" minOccurs="0"/>
      <xsd:element name="upload" type="wd:upload" minOccurs="0"/>
      <xsd:element name="alarm" type="wd:alarm" minOccurs="0"/>
      <xsd:element name="scheduler" type="wd:scheduler"
minOccurs="0"/>
      <xsd:element name="wavenis" type="wd:wavenis" minOccurs="0"/>
      <xsd:element name="metering" type="wd:metering"
minOccurs="0"/>
      <xsd:element name="rfid" type="wd:rfid" minOccurs="0"/>
      <xsd:element name="modbus" type="wd:modbus" minOccurs="0"/>
      <xsd:element name="system" type="wd:system" minOccurs="0"/>
    </xsd:sequence>
    <!-- @partial deprecated by @factory -->
    <xsd:attribute name="partial" type="xsd:boolean" default="true"
use="optional"/>
    <xsd:attribute name="factory" type="xsd:boolean" default="false"
use="optional"/>
  </xsd:complexType>
  <xsd:key name="schedule_id">
    <xsd:selector xpath="./scheduler/schedules/schedule"/>
    <xsd:field xpath="id"/>
  </xsd:key>
  <xsd:keyref name="upload_schedule_id_ref" refer="wd:schedule_id">
    <xsd:selector xpath="./upload/data"/>
    <xsd:field xpath="schedule"/>
  </xsd:keyref>
  <xsd:keyref name="wavenis_schedule_id_ref" refer="wd:schedule_id">
    <xsd:selector xpath="./wavenis/modules/module"/>
```

```
    <xsd:field xpath="schedule"/>
  </xsd:keyref>
  <xsd:keyref name="follow_schedule_id_ref" refer="wd:schedule_id">
    <xsd:selector xpath="./scheduler/schedules/schedule"/>
    <xsd:field xpath="parent"/>
  </xsd:keyref>
  <xsd:keyref name="pulse_schedule_id_ref" refer="wd:schedule_id">
    <xsd:selector xpath="./metering/pulse"/>
    <xsd:field xpath="schedule"/>
  </xsd:keyref>
  <xsd:keyref name="mbus_schedule_id_ref" refer="wd:schedule_id">
    <xsd:selector xpath="./metering/mbus"/>
    <xsd:field xpath="schedule"/>
  </xsd:keyref>
</xsd:element>

<xsd:complexType name='com'>
  <xsd:sequence>
    <xsd:element name="modem" type="wd:modem" minOccurs="0"/>
    <xsd:element name="ethernet" type="wd:ethernet" minOccurs="0"/>
    <xsd:element name="keepalive" type="wd:keepalive"
minOccurs="0"/>
    <xsd:element name="request" type="wd:request" minOccurs="0"/>
    <xsd:element name="time" type="wd:time" minOccurs="0"/>
    <xsd:element name="ftp" type="wd:ftp" minOccurs="0"/>
    <xsd:element name="ws" type="wd:ws" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='modem'>
  <xsd:sequence>
    <xsd:element name="pin" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="mode">
            <xsd:simpleType>
              <xsd:restriction base="xsd:string">
                <xsd:enumeration value="off"/>
                <xsd:enumeration value="manual"/>
                <xsd:enumeration value="automatic"/>
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
          <xsd:element name="code" type="wd:pin_code"
minOccurs="0"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
```

```
    <xsd:element name="call_number" type="xsd:string"
minOccurs="0"/>
    <xsd:element name="apn" type="xsd:string" minOccurs="0"/>
    <xsd:element name="login" type="xsd:string" minOccurs="0"/>
    <xsd:element name="password" type="xsd:string" minOccurs="0"/>
    <xsd:element name="mode" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="ondemand"/>
          <xsd:enumeration value="alwayson"/>
          <xsd:enumeration value="alwaysoff"/>
          <xsd:enumeration value="off"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="delay" type="xsd:integer" minOccurs="0"/>
    <xsd:element name="whitelist" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="caller_id"
            minOccurs='0' maxOccurs='unbounded' type="xsd:string"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='ethernet'>
  <xsd:sequence>
    <xsd:element name="use_dhcp" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="ip" type="wd:ipv4" minOccurs="0"/>
    <xsd:element name="netmask" type="wd:ipv4" minOccurs="0"/>
    <xsd:element name="gateway" minOccurs="0">
      <xsd:simpleType>
        <xsd:union memberTypes="wd:ipv4 wd:empty"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="dns" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="server"
            minOccurs='0' maxOccurs='unbounded' type="wd:ipv4"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
```

```
<xsd:complexType name='keepalive'>
  <xsd:sequence>
    <xsd:element name="method" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="icmp"/>
          <xsd:enumeration value="tcp"/>
          <xsd:enumeration value="off"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="address" type="xsd:string" minOccurs="0"/>
    <xsd:element name="port" type="wd:port" minOccurs="0"/>
    <xsd:element name="period" type="xsd:positiveInteger"
minOccurs="0"/>
    <xsd:element name="timeout" type="xsd:positiveInteger"
minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='request'>
  <xsd:sequence>
    <xsd:element name="upload" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="include_status" type="xsd:boolean"
minOccurs="0"/>
    <xsd:element name="sms_status_recipient" type="xsd:string"
minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='time'>
  <xsd:sequence>
    <xsd:element name="ntp" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="server"
            minOccurs='0' maxOccurs='unbounded' type="xsd:string"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="timezone" type="xsd:string" minOccurs="0"/>
    <xsd:element name="alarm_threshold" type="xsd:integer"
minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='ftp'>
  <xsd:sequence>
```

```
    <xsd:element name="address" type="xsd:string" minOccurs="0"/>
    <xsd:element name="login" type="xsd:string" minOccurs="0"/>
    <xsd:element name="password" type="xsd:string" minOccurs="0"/>
    <xsd:element name="mode" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="passive"/>
          <xsd:enumeration value="active"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="secured" type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="root_path" type="xsd:string" minOccurs="0"/>
    <xsd:element name="ws_notification" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="none"/>
          <xsd:enumeration value="put"/>
          <xsd:enumeration value="get"/>
          <xsd:enumeration value="both"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='ws'>
  <xsd:sequence>
    <xsd:element name="address" type="xsd:string" minOccurs="0"/>
    <xsd:element name="login" type="xsd:string" minOccurs="0"/>
    <xsd:element name="password" type="xsd:string" minOccurs="0"/>
    <xsd:element name="secured" type="xsd:boolean" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='upload'>
  <xsd:sequence>
    <xsd:element name="config" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="method" type="wd:upload_method_none"
minOccurs="0"/>
          <xsd:element name="omit_password" type="xsd:boolean"
minOccurs="0"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="supervision" minOccurs="0">
```

```
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="method" type="wd:upload_method"
minOccurs="0"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="alarm" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="method" type="wd:upload_method"
minOccurs="0"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="data" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="method" type="wd:upload_method"
minOccurs="0"/>
          <xsd:element name="format" type="wd:upload_format"
minOccurs="0"/>
          <xsd:element name="schedule" minOccurs="0">
            <xsd:simpleType>
              <xsd:union memberTypes="xsd:positiveInteger
wd:empty"/>
            </xsd:simpleType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="common" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="size_limit" type="xsd:positiveInteger"
minOccurs="0"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="upload_method">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="ftp"/>
    <xsd:enumeration value="ws"/>
  </xsd:restriction>
</xsd:simpleType>
```

```
<xsd:simpleType name="upload_method_none">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="none"/>
    <xsd:enumeration value="ftp"/>
    <xsd:enumeration value="ws"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="upload_format">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="xml"/>
    <xsd:enumeration value="csv"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name='alarm'>
  <xsd:sequence>
    <xsd:element name="sources" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="power" type="wd:alarm_mode"
            minOccurs="0"/>
          <xsd:element name="modem_ip" type="wd:alarm_mode"
            minOccurs="0"/>
          <xsd:element name="msisdn" type="wd:alarm_mode"
            minOccurs="0"/>
          <xsd:element name="sw_version" type="wd:alarm_mode"
            minOccurs="0"/>
          <xsd:element name="defaults">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="ignored" type="xsd:string"
minOccurs="0"/>
                <xsd:element name="delayed" type="xsd:string"
minOccurs="0"/>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="d_inputs" type="wd:d_inputs"
            minOccurs="0"/>
          <xsd:element name="d_output" type="wd:d_output"
            minOccurs="0"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
```

```
<xsd:simpleType name="alarm_mode">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="on"/>
    <xsd:enumeration value="off"/>
    <xsd:enumeration value="delayed"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="io_alarm_type">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="none"/>
    <xsd:enumeration value="raising"/>
    <xsd:enumeration value="falling"/>
    <xsd:enumeration value="both"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name='d_inputs'>
  <xsd:sequence>
    <xsd:element name="d_input" minOccurs="0" maxOccurs="unbounded">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="index" type="xsd:integer"/>
          <xsd:element name="label" type="xsd:string"
minOccurs="0"/>
          <xsd:element name="mode" type="wd:alarm_mode"
minOccurs="0"/>
          <xsd:element name="type" type="wd:io_alarm_type"
minOccurs="0"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='d_output'>
  <xsd:sequence>
    <xsd:element name="label" type="xsd:string" minOccurs="0"/>
    <xsd:element name="mode" type="wd:alarm_mode" minOccurs="0"/>
    <xsd:element name="type" type="wd:io_alarm_type" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='scheduler'>
  <xsd:sequence>
    <xsd:element name="schedules" minOccurs="0">
      <xsd:complexType>
```

```
        <xsd:sequence>
          <xsd:element name="schedule" type="wd:schedule"
            minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='schedule'>
  <xsd:sequence>
    <xsd:element name="id" type="xsd:positiveInteger"/>
    <xsd:element name="label" type="xsd:string"/>
    <xsd:element name="type">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="day"/>
          <xsd:enumeration value="week"/>
          <xsd:enumeration value="month"/>
          <xsd:enumeration value="year"/>
          <xsd:enumeration value="follow"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="parent" minOccurs="0">
      <xsd:simpleType>
        <xsd:union memberTypes="xsd:positiveInteger wd:empty"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="start" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="time" minOccurs="0">
            <xsd:simpleType>
              <xsd:union memberTypes="xsd:time wd:empty"/>
            </xsd:simpleType>
          </xsd:element>
          <xsd:element name="datetime" minOccurs="0">
            <xsd:simpleType>
              <xsd:union memberTypes="xsd:dateTime wd:empty"/>
            </xsd:simpleType>
          </xsd:element>
          <xsd:element name="dayofweek" minOccurs="0">
            <xsd:simpleType>
              <xsd:union memberTypes="wd:schedule_dayofweek
wd:empty"/>
            </xsd:simpleType>
          </xsd:element>
```

```
            <xsd:element name="dayofmonth" minOccurs="0">
              <xsd:simpleType>
                <xsd:union memberTypes="wd:schedule_dayofmonth
wd:empty"/>
              </xsd:simpleType>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="interval" minOccurs="0">
        <xsd:simpleType>
          <xsd:union memberTypes="xsd:positiveInteger wd:empty"/>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="count" minOccurs="0">
        <xsd:simpleType>
          <xsd:union memberTypes="xsd:positiveInteger wd:empty"/>
        </xsd:simpleType>
      </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name='schedule_dayofweek'>
  <xsd:restriction base="xsd:integer">
    <xsd:minInclusive value="1"/>
    <xsd:maxInclusive value="7"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name='schedule_dayofmonth'>
  <xsd:restriction base="xsd:integer">
    <xsd:minInclusive value="1"/>
    <xsd:maxInclusive value="31"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name='wavenis'>
  <xsd:sequence>
    <xsd:element name="bridge" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="enabled" type="xsd:boolean"
minOccurs="0"/>
          <xsd:element name="port" type="wd:port" minOccurs="0"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="time" minOccurs="0">
```

```
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="mode" minOccurs="0">
            <xsd:simpleType>
              <xsd:restriction base="xsd:string">
                <xsd:enumeration value="utc"/>
                <xsd:enumeration value="local"/>
                <xsd:enumeration value="nodst"/>
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="alarm" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="mode" minOccurs="0">
            <xsd:simpleType>
              <xsd:restriction base="xsd:string">
                <xsd:enumeration value="basic"/>
                <xsd:enumeration value="extended"/>
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
          <xsd:element name="sources" minOccurs="0">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="unknown"
                    minOccurs="0" type="wd:alarm_mode"/>
                <xsd:element name="route"
                    minOccurs="0" type="wd:alarm_mode"/>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="modules" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="module" type="wd:wavenis_module"
minOccurs="0"
            maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
```

```
</xsd:complexType>

<xsd:complexType name='wavenis_module'>
  <xsd:sequence>
    <xsd:element name="address" type="wd:wavenis_address"/>
    <xsd:element name="label" type="xsd:string"/>
    <xsd:element name="type">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="waveflow"/>
          <xsd:enumeration value="wavetherm_dallas"/>
          <xsd:enumeration value="wavetherm_pt100"/>
          <xsd:enumeration value="wavetherm_pt1000"/>
          <xsd:enumeration value="wavesense_4-20ma"/>
          <xsd:enumeration value="wavesense_0-5v"/>
          <xsd:enumeration value="wavelog"/>
          <xsd:enumeration value="wavetic"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="repeaters" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="repeater"
            minOccurs='0' maxOccurs='3'
            type="wd:wavenis_address"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="mode">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="immediate"/>
          <xsd:enumeration value="datalog"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="nbinput">
      <xsd:simpleType>
        <xsd:restriction base="xsd:integer">
          <xsd:minInclusive value="1"/>
          <xsd:maxInclusive value="4"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="schedule" type="xsd:positiveInteger"/>
  </xsd:sequence>
</xsd:complexType>
```

```
<xsd:complexType name='metering'>
  <xsd:sequence>
    <xsd:element name="pulse" type="wd:pulse_config" minOccurs="0"/>
    <xsd:element name="mbus" type="wd:mbus_config" minOccurs="0"/>
    <xsd:element name="wmbus" type="wd:wmbus_config" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='pulse_config'>
  <xsd:sequence>
    <xsd:element name="schedule" minOccurs="0">
      <xsd:simpleType>
        <xsd:union memberTypes="xsd:positiveInteger wd:empty"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="input_1" type="wd:pulse_input"
minOccurs="0"/>
    <xsd:element name="input_2" type="wd:pulse_input"
minOccurs="0"/>
    <xsd:element name="input_3" type="wd:pulse_input"
minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='pulse_input'>
  <xsd:sequence>
    <xsd:element name="label" type="xsd:string"/>
    <xsd:element name="unit" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='mbus_config'>
  <xsd:sequence>
    <xsd:element name="schedule">
      <xsd:simpleType>
        <xsd:union memberTypes="xsd:positiveInteger wd:empty"/>
      </xsd:simpleType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='wmbus_config'>
  <xsd:sequence>
    <xsd:element name="mode" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="T1"/>
```

```
            <xsd:enumeration value="S1"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="long_preamble" type="xsd:boolean"
minOccurs="0"/>
      <xsd:element name="modules" minOccurs="0">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="module" type="wd:wmbus_module"
minOccurs="0"
              maxOccurs="unbounded" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='wmbus_module'>
  <xsd:sequence>
    <xsd:element name="address" type="xsd:string"/>
    <xsd:element name="label" type="xsd:string"/>
    <xsd:element name="key" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='system'>
  <xsd:sequence>
    <xsd:element name="log" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="level" type="xsd:string"
minOccurs="0"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="password" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="admin" type="wd:password"
minOccurs="0"/>
          <xsd:element name="install" type="wd:password"
minOccurs="0"/>
          <xsd:element name="data" type="wd:password"
minOccurs="0"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
```

```
    <xsd:element name="ports" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="rs232" minOccurs="0">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="mode">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:enumeration value="off"/>
                            <xsd:enumeration value="mbus"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:element>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="rs485" minOccurs="0">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="mode" minOccurs="0">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:enumeration value="off"/>
                            <xsd:enumeration value="modbus"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:element>
                <xsd:element name="baudrate"
type="wd:uart_baudrate" minOccurs="0"/>
                <xsd:element name="data" type="wd:uart_data_bits"
minOccurs="0"/>
                <xsd:element name="parity" type="wd:uart_parity"
minOccurs="0"/>
                <xsd:element name="stop_bit"
type="wd:uart_stop_bits" minOccurs="0"/>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="input_1" minOccurs="0">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="mode">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:enumeration value="d_input"/>
                            <xsd:enumeration value="pulse"/>
                        </xsd:restriction>
```

```
                    </xsd:simpleType>
                  </xsd:element>
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
            <xsd:element name="input_2" minOccurs="0">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="mode">
                      <xsd:simpleType>
                           <xsd:restriction base="xsd:string">
                                <xsd:enumeration value="d_input"/>
                                <xsd:enumeration value="pulse"/>
                           </xsd:restriction>
                      </xsd:simpleType>
                  </xsd:element>
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
            <xsd:element name="input_3" minOccurs="0">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="mode">
                      <xsd:simpleType>
                           <xsd:restriction base="xsd:string">
                                <xsd:enumeration value="d_input"/>
                                <xsd:enumeration value="pulse"/>
                           </xsd:restriction>
                      </xsd:simpleType>
                  </xsd:element>
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='rfid'>
  <xsd:sequence>
    <xsd:element name="rssi_threshold" type="xsd:integer"
minOccurs="0" />
    <xsd:element name="crc" type="xsd:integer" minOccurs="0" />
    <xsd:element name="detection_delay" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="entering" type="xsd:integer"
```

```
minOccurs="0" />
          <xsd:element name="leaving" type="xsd:integer"
minOccurs="0" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="alarm" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="sources" minOccurs="0">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="entering" type="wd:alarm_mode"
minOccurs="0" />
                <xsd:element name="leaving" type="wd:alarm_mode"
minOccurs="0" />
                <xsd:element name="id_flags" type="wd:alarm_mode"
minOccurs="0" />
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='modbus'>
  <xsd:sequence>
    <xsd:element name="tcp" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="timeout" type="xsd:integer"
minOccurs="0" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="rtu" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="timeout" type="xsd:integer"
minOccurs="0" />
          <xsd:element name="turnaround" type="xsd:integer"
minOccurs="0" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="datasets" type="wd:modbus_datasets"
```

```
minOccurs="0" />
    <xsd:element name="modules" type="wd:modbus_modules"
minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='modbus_datasets'>
  <xsd:sequence>
    <xsd:element name="dataset" minOccurs="0" maxOccurs="unbounded">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="id" type="xsd:integer"/>
          <xsd:element name="label" type="xsd:string" minOccurs="0"
/>
          <xsd:element name="vars" type="wd:modbus_vars"
minOccurs="0" />
          <xsd:element name="boundaries"
type="wd:modbus_boundaries" minOccurs="0" />
          <xsd:element name="polling" type="xsd:boolean"
minOccurs="0" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='modbus_vars'>
  <xsd:sequence>
    <xsd:element name="var" minOccurs="0" maxOccurs="unbounded">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="name" type="xsd:string"/>
          <xsd:element name="type" type="wd:modbus_type"/>
          <xsd:element name="address" type="wd:hex_string"/>
          <xsd:element name="size" type="xsd:integer" minOccurs="0"
/>
          <xsd:element name="format" type="wd:modbus_format"/>
          <xsd:element name="flags" type="wd:modbus_flags"
minOccurs="0"/>
          <xsd:element name="threshold" minOccurs="0" >
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="low">
                    <xsd:simpleType>
                        <xsd:union memberTypes="xsd:string
wd:empty"/>
                    </xsd:simpleType>
                </xsd:element>
```

```
                    <xsd:element name="high">
                        <xsd:simpleType>
                            <xsd:union memberTypes="xsd:string
wd:empty"/>
                        </xsd:simpleType>
                    </xsd:element>
                    <xsd:element name="hysteresis">
                        <xsd:simpleType>
                            <xsd:union memberTypes="xsd:string
wd:empty"/>
                        </xsd:simpleType>
                    </xsd:element>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:complexType>



<xsd:complexType name='modbus_boundaries'>
  <xsd:sequence>
    <xsd:element name="boundary" minOccurs="0"
maxOccurs="unbounded">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="type" type="wd:modbus_type"/>
          <xsd:element name="address" type="wd:hex_string"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='modbus_modules'>
  <xsd:sequence>
    <xsd:element name="module" minOccurs="0" maxOccurs="unbounded">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="label" type="xsd:string"
minOccurs="0"/>
          <xsd:element name="dataset" type="xsd:integer"/>
          <xsd:element name="address" type="wd:modbus_addr"
minOccurs="0"/>
          <xsd:element name="ip" minOccurs="0">
```

```
            <xsd:simpleType>
              <xsd:union memberTypes="wd:ipv4 wd:empty"/>
            </xsd:simpleType>
          </xsd:element>
          <xsd:element name="schedule" type="xsd:integer"
minOccurs="0"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name='modbus_addr'>
  <xsd:restriction base="xsd:integer">
    <xsd:minInclusive value="1"/>
    <xsd:maxInclusive value="247"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name='modbus_type'>
  <xsd:restriction base='xsd:string'>
    <xsd:pattern value='S[0,1,3,4]'/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="modbus_format">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="raw"/>
    <xsd:enumeration value="boolean"/>
    <xsd:enumeration value="integer"/>
    <xsd:enumeration value="float"/>
    <xsd:enumeration value="ascii"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="modbus_flags">
        <xsd:restriction base="xsd:string">
<!--
                <xsd:enumeration value="cmd_only"/>
                <xsd:enumeration value="little_endian"/>
                <xsd:enumeration value="no_opt"/>
                <xsd:enumeration value="is_status"/>
                <xsd:enumeration value="is_alarm"/>
                <xsd:enumeration value="signed"/>
                <xsd:enumeration value=""/>
-->
        </xsd:restriction>
</xsd:simpleType>
```

```
<xsd:simpleType name='wavenis_address'>
  <xsd:restriction base='xsd:string'>
    <xsd:pattern value='([0-9a-zA-Z]{12})|([0-9]{5}-?[0-9]{2}-?[0-
9]{8})'/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name='pin_code'>
  <xsd:restriction base='xsd:string'>
    <xsd:pattern value='[0-9]{4,6}'/>
  </xsd:restriction>
</xsd:simpleType>

<!--
password:
  ASCII characters from 0x21 to 0x7E
  Except: " (0x22), & (0x26), ' (0x27), < (0x3C), > (0x3E), ?
(0x3F), ` (0x60)
-->
<xsd:simpleType name='password'>
  <xsd:restriction base='xsd:string'>
    <xsd:pattern value='[&#x0021;-&#x007E;-
[&#x0022;&#x0026;&#x0027;&#x003C;&#x003E;&#x003F;&#x0060;]]{1,120}'/
>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="empty">
  <xsd:restriction base="xsd:string">
    <xsd:length value="0"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name='ipv4'>
  <xsd:restriction base='xsd:string'>
    <xsd:pattern value='[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-
9]{1,3}'/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name='port'>
  <xsd:restriction base="xsd:integer">
    <xsd:minInclusive value="1"/>
    <xsd:maxInclusive value="65535"/>
  </xsd:restriction>
</xsd:simpleType>
```

```
<xsd:simpleType name='uart_baudrate'>
  <xsd:restriction base='xsd:integer'>
    <xsd:pattern
value='1200|2400|4800|9600|19200|38400|57600|115200'/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name='uart_data_bits'>
  <xsd:restriction base='xsd:string'>
    <xsd:pattern value='[5-9]'/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="uart_parity">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="odd"/>
    <xsd:enumeration value="even"/>
    <xsd:enumeration value="none"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name='uart_stop_bits'>
  <xsd:restriction base='xsd:string'>
    <xsd:pattern value='[1-2]'/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name='hex_string'>
        <xsd:restriction base='xsd:string'>
                <xsd:pattern value='[0-9a-zA-Z]*'/>
        </xsd:restriction>
</xsd:simpleType>

</xsd:schema>
```

19 ANNEXE B Exemple XML – Configuration

config.xml

```
<?xml version="1.0"?>
<config
  xmlns="http://www.webdyn.com/WRF_config_20110606"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.webdyn.com/WRF_config_20110606
config.xsd">
  <uid>07F38D</uid>
  <name>WGRF_07F38D</name>
  <enable_local_config>true</enable_local_config>
  <com>
    <modem>
      <pin>
        <mode>manual</mode>
        <code>1234</code>
      </pin>
      <call_number>*99***1#</call_number>
      <apn>m2minternet</apn>
      <login>login</login>
      <password>password</password>
      <mode>ondemand</mode>
      <whitelist>
        <caller_id>+33123242526</caller_id>
        <caller_id>+33123242527</caller_id>
      </whitelist>
    </modem>
    <ethernet>
      <use_dhcp>false</use_dhcp>
      <ip>192.168.10.10</ip>
      <netmask>255.255.255.0</netmask>
      <gateway>192.168.10.254</gateway>
      <dns>
        <server>192.168.10.254</server>
        <server>8.8.8.8</server>
        <server>8.8.4.4</server>
      </dns>
    </ethernet>
    <keepalive>
      <method>tcp</method>
      <address>12.13.14.15</address>
      <port>8003</port>
      <period>1800</period>
      <timeout>10</timeout>
```

```
    </keepalive>
    <request>
      <upload>true</upload>
      <include_status>true</include_status>
      <sms_status_recipient/>
    </request>
    <time>
      <ntp>
        <server>pool.ntp.org</server>
      </ntp>
      <timezone>Europe/Paris</timezone>
      <alarm_threshold>300</alarm_threshold>
    </time>
    <ftp>
      <address>12.13.14.16</address>
      <login>login</login>
      <password>password</password>
      <mode>passive</mode>
      <secured>false</secured>
      <root_path>/</root_path>
      <ws_notification>none</ws_notification>
    </ftp>
    <ws>
      <address>12.13.14.16</address>
      <login>login</login>
      <password>password</password>
      <secured>false</secured>
    </ws>
  </com>
  <upload>
    <config>
      <method>ftp</method>
    </config>
    <supervision>
      <method>ftp</method>
    </supervision>
    <alarm>
      <method>ws</method>
    </alarm>
    <data>
      <method>ftp</method>
      <format>csv</format>
      <schedule>1</schedule>
    </data>
  </upload>
  <alarm>
    <sources>
      <defaults>
```

```
        <ignored/>
        <delayed/>
      </defaults>
      <d_inputs>
        <d_input>
          <index>0</index>
          <label>Intrusion dectection</label>
          <type>both</type>
        </d_input>
      </d_inputs>
    </sources>
  </alarm>
  <scheduler>
    <schedules>
      <schedule>
        <id>1</id>
        <label>upload schedule</label>
        <type>week</type>
        <start>
          <time>01:00:00</time>
          <dayofweek>7</dayofweek>
        </start>
        <interval>1</interval>
        <count>1</count>
      </schedule>
      <schedule>
        <id>2</id>
        <label>collect schedule</label>
        <type>day</type>
        <start>
          <time>04:00:00</time>
        </start>
        <interval>43200</interval>
        <count>2</count>
      </schedule>
    </schedules>
  </scheduler>
  <wavenis>
    <bridge>
      <enabled>true</enabled>
      <port>4000</port>
    </bridge>
    <modules>
      <module>
        <address>011A1030A5D4</address>
        <label>main</label>
        <type>waveflow</type>
        <repeaters>
```

```
            <repeater>011A1030A7D3</repeater>
          </repeaters>
          <mode>datalog</mode>
          <nbinput>4</nbinput>
          <schedule>2</schedule>
        </module>
      </modules>
    </wavenis>
    <metering>
      <pulse>
        <schedule/>
        <input_1>
          <label/>
          <unit/>
        </input_1>
        <input_2>
          <label/>
          <unit/>
        </input_2>
        <input_3>
          <label/>
          <unit/>
        </input_3>
      </pulse>
      <mbus>
        <schedule/>
      </mbus>
      <wmbus>
        <mode>T1</mode>
        <long_preamble>false</long_preamble>
        <modules>
          <module>
            <address>012345678901</address>
            <label>test1</label>
            <key>00112233445566778899</key>
          </module>
        </modules>
      </wmbus>
    </metering>
    <rfid>
      <rssi_threshold>255</rssi_threshold>
      <crc>0</crc>
      <detection_delay>
        <entering>600</entering>
        <leaving>500</leaving>
      </detection_delay>
      <alarm>
        <sources>
```

```
        <entering>on</entering>
        <leaving>on</leaving>
        <id_flags>on</id_flags>
      </sources>
    </alarm>
  </rfid>
  <modbus>
    <rtu>
      <timeout>500</timeout>
      <turnaround>20</turnaround>
    </rtu>
    <datasets>
      <dataset>
        <id>1</id>
        <label>dataset_1</label>
        <vars>
          <var>
            <name>var1</name>
            <type>S4</type>
            <address>0x1234</address>
            <size>2</size>
            <format>integer</format>
            <flags>is_alarm</flags>
          </var>
          <var>
            <name>var1</name>
            <type>S4</type>
            <address>0x1234</address>
            <size>2</size>
            <format>integer</format>
            <flags></flags>
          </var>
          <var>
            <name>var1</name>
            <type>S4</type>
            <address>0x1234</address>
            <size>2</size>
            <format>integer</format>
            <flags>is_alarm,is_status</flags>
          </var>
          <var>
            <name>var2</name>
            <type>S0</type>
            <address>1234</address>
            <size>1</size>
            <format>boolean</format>
            <flags>is_status</flags>
            <threshold>
```

```
                <low>20</low>
                <high>40</high>
                <hysteresis>2</hysteresis>
              </threshold>
            </var>
          </vars>
          <boundaries>
            <boundary>
              <type>S0</type>
              <address>1234</address>
            </boundary>
          </boundaries>
          <polling>true</polling>
        </dataset>
      </datasets>
      <modules>
        <module>
          <label>my_module</label>
          <dataset>1</dataset>
          <address>26</address>
          <schedule>1</schedule>
        </module>
      </modules>
    </modbus>
</config>
```

20 ANNEXE C Schéma XSD – Alarmes

alarm.xsd

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:wd="http://www.webdyn.com/WRF_alarm_20110606"
  targetNamespace="http://www.webdyn.com/WRF_alarm_20110606"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

<xsd:element name="alarms">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="uid" type="xsd:string"/>
      <xsd:choice maxOccurs="unbounded">
        <xsd:element name="command" type="wd:cmd_alarm"/>
        <xsd:element name="ntp" type="wd:ntp_alarm"/>
        <xsd:element name="sw_version" type="wd:version_alarm"/>
        <xsd:element name="msisdn" type="wd:msisdn_alarm"/>
        <xsd:element name="power" type="wd:power_alarm"/>
        <xsd:element name="modem_ip" type="wd:ip_alarm"/>
        <xsd:element name="default" type="wd:default_alarm"/>
        <xsd:element name="d_input" type="wd:d_input_alarm"/>
        <xsd:element name="d_output" type="wd:d_output_alarm"/>
        <xsd:element name="wavenis_unknown"
type="wd:wavenis_topo_alarm"/>
        <xsd:element name="wavenis_route"
type="wd:wavenis_topo_alarm"/>
        <xsd:element name="wavenis_cmd"
type="wd:wavenis_cmd_alarm"/>
        <xsd:element name="waveflow" type="wd:waveflow_alarm"/>
        <xsd:element name="wavetherm"
type="wd:wavethermsense_alarm"/>
        <xsd:element name="wavesense"
type="wd:wavethermsense_alarm"/>
        <xsd:element name="wavelog" type="wd:wavelog_alarm"/>
        <xsd:element name="wavetic" type="wd:wavetic_alarm"/>
        <xsd:element name="wavetalk" type="wd:wavetalk_alarm"/>
        <xsd:element name="modbus_cmd" type="wd:modbus_cmd_alarm"/>
        <xsd:element name="modbus" type="wd:modbus"/>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

```
<xsd:complexType name='alarm' abstract="true">
  <xsd:sequence>
    <xsd:element name="date" type="xsd:dateTime"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='cmd_alarm_base' abstract="true">
        <xsd:complexContent>
                <xsd:extension base="wd:alarm">
                        <xsd:sequence>
                                <xsd:element name="cid"
type="xsd:string" minOccurs="0"/>
                                <xsd:element name="source">
                                        <xsd:simpleType>
                                                <xsd:restriction
base="xsd:string">

<xsd:enumeration value="sms"/>

<xsd:enumeration value="ws"/>

<xsd:enumeration value="ftp"/>

<xsd:enumeration value="local"/>
                                                </xsd:restriction>
                                        </xsd:simpleType>
                                </xsd:element>
                        </xsd:sequence>
                </xsd:extension>
        </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="cmd_alarm">
  <xsd:complexContent>
    <xsd:extension base="wd:cmd_alarm_base">
      <xsd:sequence>
        <xsd:element name="error">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="invalid_uid"/>
              <xsd:enumeration value="malformed"/>
              <xsd:enumeration value="other"/>
              <xsd:enumeration value="none"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="description" type="xsd:string"
minOccurs="0"/>
```

```
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ntp_alarm">
  <xsd:complexContent>
    <xsd:extension base="wd:alarm">
      <xsd:sequence>
        <xsd:element name="before" type="xsd:dateTime"/>
        <xsd:element name="after" type="xsd:dateTime"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="power_alarm">
  <xsd:complexContent>
    <xsd:extension base="wd:alarm">
      <xsd:sequence>
        <xsd:element name="state">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="on"/>
              <xsd:enumeration value="off"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="version_alarm">
  <xsd:complexContent>
    <xsd:extension base="wd:alarm">
      <xsd:sequence>
        <xsd:element name="version" type="xsd:string"/>
        <xsd:element name="kernel" type="xsd:string"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="msisdn_alarm">
  <xsd:complexContent>
    <xsd:extension base="wd:alarm">
      <xsd:sequence>
```

```
        <xsd:element name="msisdn" type="xsd:string"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ip_alarm">
  <xsd:complexContent>
    <xsd:extension base="wd:alarm">
      <xsd:sequence>
        <xsd:element name="ip" type="wd:ipv4"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="default_alarm">
  <xsd:complexContent>
    <xsd:extension base="wd:alarm">
      <xsd:sequence>
        <xsd:element name="name" type="xsd:string"/>
        <xsd:element name="type" type="wd:basic_alarm_type"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="d_input_alarm">
  <xsd:complexContent>
    <xsd:extension base="wd:alarm">
      <xsd:sequence>
        <xsd:element name="index" type="xsd:integer"/>
        <xsd:element name="label" type="xsd:string"/>
        <xsd:element name="type" type="wd:basic_alarm_type"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="d_output_alarm">
  <xsd:complexContent>
    <xsd:extension base="wd:alarm">
      <xsd:sequence>
        <xsd:element name="label" type="xsd:string"/>
        <xsd:element name="type" type="wd:basic_alarm_type"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
```

```
</xsd:complexType>

<xsd:simpleType name='basic_alarm_type'>
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="set"/>
    <xsd:enumeration value="reset"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="wavenis_topo_alarm">
  <xsd:complexContent>
    <xsd:extension base="wd:alarm">
      <xsd:sequence>
        <xsd:element name="address" type="wd:wavenis_address"/>
        <xsd:element name="repeaters" minOccurs="0">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="repeater"
                minOccurs='0' maxOccurs='3'
                type="wd:wavenis_address"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="wavenis_cmd_alarm">
  <xsd:complexContent>
    <xsd:extension base="wd:cmd_alarm_base">
      <xsd:sequence>
        <xsd:element name="subcmd" type="xsd:string"/>
        <xsd:element name="address" type="wd:wavenis_address"/>
        <xsd:element name="result">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="ok"/>
              <xsd:enumeration value="error"/>
              <xsd:enumeration value="unsupported"/>
              <xsd:enumeration value="open"/>
              <xsd:enumeration value="close"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="request" type="wd:hex_string"
minOccurs="0"/>
        <xsd:element name="response" type="wd:hex_string"
```

```
minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name='wavetic_alarm'>
  <xsd:complexContent>
    <xsd:extension base="wd:alarm">
      <xsd:sequence>
        <xsd:element name="address" type="wd:wavenis_address"/>
        <xsd:element name="eol_battery" type="xsd:boolean"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name='wavetalk_alarm'>
  <xsd:complexContent>
    <xsd:extension base="wd:alarm">
      <xsd:sequence>
        <xsd:element name="address" type="wd:wavenis_address"/>
        <xsd:element name="life-counter" type="xsd:integer"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name='wavethermsense_alarm'>
  <xsd:complexContent>
    <xsd:extension base="wd:alarm">
      <xsd:sequence>
        <xsd:element name="address" type="wd:wavenis_address"/>
        <xsd:element name="input" type="wd:wavenis_input"
minOccurs="0"/>
        <xsd:element name="threshold">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="high" type="xsd:boolean"/>
              <xsd:element name="low" type="xsd:boolean"/>
              <xsd:element name="duration" type="xsd:integer"
                minOccurs="0"/>
              <xsd:element name="value" type="xsd:string"
                minOccurs="0"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="sensor_fault" type="xsd:boolean"/>
```

```
        <xsd:element name="eol_battery" type="xsd:boolean"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name='wavelog_alarm'>
  <xsd:complexContent>
    <xsd:extension base="wd:alarm">
      <xsd:sequence>
        <xsd:element name="address" type="wd:wavenis_address"/>
        <xsd:element name="app_status">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="reset_flag" type="xsd:boolean"/>
              <xsd:element name="input_conflict"
type="xsd:boolean"/>
              <xsd:element name="discontinuity"
type="xsd:boolean"/>
              <xsd:element name="eol_battery" type="xsd:boolean"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="io_status">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="input1" type="xsd:boolean"/>
              <xsd:element name="input2" type="xsd:boolean"/>
              <xsd:element name="input3" type="xsd:boolean"/>
              <xsd:element name="input4" type="xsd:boolean"/>
              <xsd:element name="output1" type="xsd:boolean"/>
              <xsd:element name="output2" type="xsd:boolean"/>
              <xsd:element name="output3" type="xsd:boolean"/>
              <xsd:element name="output4" type="xsd:boolean"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="open" type="xsd:boolean"/>
        <xsd:element name="close" type="xsd:boolean"/>
        <xsd:element name="open_timeout" type="xsd:boolean"/>
        <xsd:element name="close_timeout" type="xsd:boolean"/>
        <xsd:element name="input1" type="xsd:boolean"/>
        <xsd:element name="input2" type="xsd:boolean"/>
        <xsd:element name="input3" type="xsd:boolean"/>
        <xsd:element name="input4" type="xsd:boolean"/>
        <xsd:element name="eol_battery" type="xsd:boolean"/>
      </xsd:sequence>
    </xsd:extension>
```

```
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="waveflow_alarm">
  <xsd:complexContent>
    <xsd:extension base="wd:alarm">
      <xsd:sequence>
        <xsd:element name="address" type="wd:wavenis_address"/>
        <xsd:element name="type">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="basic"/>
              <xsd:enumeration value="set"/>
              <xsd:enumeration value="reset"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:choice>
          <xsd:element name="status"
type="wd:waveflow_alarm_status"/>
          <xsd:element name="leak" type='wd:wavenis_alarm_leak'/>
          <xsd:element name="backflow"
type='wd:wavenis_alarm_backflow'/>
          <xsd:element name="wire_cut"
type='wd:wavenis_alarm_input'/>
          <xsd:element name="reed_fault"
type='wd:wavenis_alarm_input'/>
          <xsd:element name="eol_battery"
type='wd:wavenis_alarm_eol'/>
        </xsd:choice>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name='waveflow_alarm_status'>
  <xsd:sequence>
    <xsd:element name="input" type="wd:wavenis_input"
minOccurs="0"/>
    <xsd:element name="leak">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="high_threshold" type="xsd:boolean"/>
          <xsd:element name="low_threshold" type="xsd:boolean"/>
          <xsd:element name="flow" type="xsd:integer"
minOccurs="0"/>
        </xsd:sequence>
      </xsd:complexType>
```

```
    </xsd:element>
    <xsd:element name="backflow" type="xsd:boolean"/>
    <xsd:element name="wire_cut" type="xsd:boolean"/>
    <xsd:element name="reed_fault" type="xsd:boolean"/>
    <xsd:element name="eol_battery" type="xsd:boolean"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='wavenis_alarm_leak'>
  <xsd:sequence>
    <xsd:element name="threshold">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="high"/>
          <xsd:enumeration value="low"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="input" type="wd:wavenis_input"/>
    <xsd:element name="flow" type="xsd:integer" minOccurs="0"/>
    <xsd:element name="date" type="xsd:dateTime"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='wavenis_alarm_backflow'>
  <xsd:sequence>
    <xsd:element name="input" type="wd:wavenis_input"/>
    <xsd:choice>
      <xsd:element name="qty_detection">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="quantity" type="xsd:integer"/>
            <xsd:element name="start" type="xsd:dateTime"/>
            <xsd:element name="end" type="xsd:dateTime"
minOccurs="0"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="flow_detection">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="flow" type="xsd:integer"/>
            <xsd:element name="duration" type="xsd:integer"/>
            <xsd:element name="end" type="xsd:dateTime"
minOccurs="0"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
```

```
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='wavenis_alarm_input'>
  <xsd:sequence>
    <xsd:element name="input" type="wd:wavenis_input"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='wavenis_alarm_eol'>
  <xsd:sequence>
    <xsd:element name="date" type="xsd:dateTime"/>
    <xsd:element name="life_counter" type="xsd:integer"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="wavenis_input">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="A"/>
    <xsd:enumeration value="B"/>
    <xsd:enumeration value="C"/>
    <xsd:enumeration value="D"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name='hex_string'>
  <xsd:restriction base='xsd:string'>
    <xsd:pattern value='[0-9a-zA-Z]*'/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name='wavenis_address'>
  <xsd:restriction base='xsd:string'>
    <xsd:pattern value='([0-9a-zA-Z]{12})|([0-9]{5}-?[0-9]{2}-?[0-
9]{8})'/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name='ipv4'>
  <xsd:restriction base='xsd:string'>
    <xsd:pattern value='[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-
9]{1,3}'/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="modbus_cmd_alarm">
       <xsd:complexContent>
```

```
                <xsd:extension base="wd:cmd_alarm_base">
                        <xsd:sequence>
                                <xsd:element name="subcmd"
type="xsd:string"/>
                                <xsd:element name="result">
                                        <xsd:simpleType>
                                                <xsd:restriction
base="xsd:string">

<xsd:enumeration value="ok"/>

<xsd:enumeration value="error"/>

<xsd:enumeration value="no_response"/>
                                                </xsd:restriction>
                                        </xsd:simpleType>
                                </xsd:element>
        <xsd:element name="description" type="xsd:string"
minOccurs="0"/>
                        </xsd:sequence>
                </xsd:extension>
        </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="modbus">
  <xsd:complexContent>
    <xsd:extension base="wd:alarm">
      <xsd:sequence>
                        <xsd:element name="module">
                                <xsd:complexType>
                                        <xsd:sequence>
                                                <xsd:element
name="label" type="xsd:string"/>
                                                 <xsd:element
name="address" type="wd:modbus_addr"/>
                                                   <xsd:element
name="ip" type="wd:ipv4" minOccurs="0"/>
                                                </xsd:sequence>
                                </xsd:complexType>
                        </xsd:element>
                          <xsd:element name="variable">
                                <xsd:complexType>
                                        <xsd:sequence>
                                                  <xsd:element
name="name" type="xsd:string"/>
                                                 <xsd:element
name="address" type="wd:hex_string"/>
                                                <xsd:element
```

```
name="type" type="wd:modbus_type"/>
                                                    </xsd:sequence>
                                                </xsd:complexType>
                                            </xsd:element>
                <xsd:element name="value" type="xsd:string" />
                <xsd:element name="status" type="xsd:string"
minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:simpleType name='modbus_addr'>
        <xsd:restriction base="xsd:integer">
                <xsd:minInclusive value="1"/>
                <xsd:maxInclusive value="247"/>
        </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name='modbus_type'>
        <xsd:restriction base='xsd:string'>
                <xsd:pattern value='S[0,1,3,4]'/>
        </xsd:restriction>
</xsd:simpleType>

</xsd:schema>
```

21 ANNEXE D Exemple XML – Alarmes

alarm.xml

```
<?xml version="1.0"?>
<alarms
  xmlns="http://www.webdyn.com/WRF_alarm_20110606"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.webdyn.com/WRF_alarm_20110606
alarm.xsd">

  <uid>07F38D</uid>

  <ntp>
    <date>2011-05-26T07:59:10</date>
    <before>2011-05-26T07:52:00</before>
    <after>2011-05-26T07:59:00</after>
  </ntp>

  <d_input>
    <date>2011-05-26T10:00:00</date>
    <index>0</index>
    <label>input_0</label>
    <type>set</type>
  </d_input>

  <d_input>
    <date>2011-05-26T10:10:00</date>
    <index>0</index>
    <label>input_0</label>
    <type>reset</type>
  </d_input>

  <d_output>
    <date>2011-05-26T10:20:00</date>
    <label>output</label>
    <type>set</type>
  </d_output>

  <power>
    <date>2011-05-26T10:30:00</date>
    <state>off</state>
  </power>

  <modem_ip>
    <date>2011-05-26T10:30:00</date>
```

```
    <ip>90.84.146.195</ip>
  </modem_ip>

  <sw_version>
    <date>2011-05-26T10:40:00</date>
    <version>v1.01</version>
    <kernel>2.6.35.6</kernel>
  </sw_version>

  <default>
    <date>2011-05-26T10:50:00</date>
    <name>D_ETHERNET</name>
    <type>set</type>
  </default>

  <default>
    <date>2011-05-26T10:55:00</date>
    <name>D_ETHERNET</name>
    <type>reset</type>
  </default>

  <waveflow>
    <date>2011-05-26T11:00:00</date>
    <address>011A1030A5D4</address>
    <type>basic</type>
    <status>
      <input>A</input>
      <leak>
        <high_threshold>true</high_threshold>
        <low_threshold>false</low_threshold>
        <flow>246</flow>
      </leak>
      <backflow>false</backflow>
      <wire_cut>false</wire_cut>
      <reed_fault>false</reed_fault>
      <eol_battery>false</eol_battery>
    </status>
  </waveflow>

  <waveflow>
    <date>2011-05-27T13:00:00</date>
    <address>00278-03-03146635</address>
    <type>set</type>
    <leak>
      <threshold>high</threshold>
      <input>A</input>
      <flow>456</flow>
      <date>2011-05-27T12:58:00</date>
```

```
    </leak>
  </waveflow>

  <waveflow>
    <date>2011-05-27T13:10:00</date>
    <address>011A1030A5D4</address>
    <type>reset</type>
    <leak>
      <threshold>high</threshold>
      <input>A</input>
      <date>2011-05-27T13:09:00</date>
    </leak>
  </waveflow>

  <waveflow>
    <date>2011-05-27T14:00:00</date>
    <address>011A1030A5D4</address>
    <type>set</type>
    <backflow>
      <input>A</input>
      <qty_detection>
        <quantity>345</quantity>
        <start>2011-05-27T13:59:00</start>
      </qty_detection>
    </backflow>
  </waveflow>

  <waveflow>
    <date>2011-05-27T14:10:00</date>
    <address>011A1030A5D4</address>
    <type>reset</type>
    <backflow>
      <input>A</input>
      <qty_detection>
        <quantity>500</quantity>
        <start>2011-05-27T13:59:00</start>
        <end>2011-05-27T14:09:00</end>
      </qty_detection>
    </backflow>
  </waveflow>

  <waveflow>
    <date>2011-05-27T15:00:00</date>
    <address>011A1030A5D4</address>
    <type>set</type>
    <backflow>
      <input>A</input>
      <flow_detection>
```

```
        <flow>323</flow>
        <duration>3</duration>
      </flow_detection>
    </backflow>
  </waveflow>

  <waveflow>
    <date>2011-05-27T15:10:00</date>
    <address>011A1030A5D4</address>
    <type>reset</type>
    <backflow>
      <input>A</input>
      <flow_detection>
        <flow>356</flow>
        <duration>13</duration>
        <end>2011-05-27T15:10:00</end>
      </flow_detection>
    </backflow>
  </waveflow>

  <waveflow>
    <date>2011-05-27T16:00:00</date>
    <address>011A1030A5D4</address>
    <type>set</type>
    <wire_cut>
      <input>B</input>
    </wire_cut>
  </waveflow>

  <waveflow>
    <date>2011-05-27T16:10:00</date>
    <address>011A1030A5D4</address>
    <type>reset</type>
    <wire_cut>
      <input>B</input>
    </wire_cut>
  </waveflow>

  <waveflow>
    <date>2011-05-27T17:00:00</date>
    <address>011A1030A5D4</address>
    <type>set</type>
    <reed_fault>
      <input>A</input>
    </reed_fault>
  </waveflow>

  <waveflow>
```

```
    <date>2011-05-27T17:10:00</date>
    <address>011A1030A5D4</address>
    <type>reset</type>
    <reed_fault>
      <input>A</input>
    </reed_fault>
  </waveflow>

  <waveflow>
    <date>2011-05-27T18:00:00</date>
    <address>011A1030A5D4</address>
    <type>set</type>
    <eol_battery>
      <date>2011-05-27T17:59:00</date>
      <life_counter>4000</life_counter>
    </eol_battery>
  </waveflow>

  <wavenis_cmd>
    <date>2011-05-27T19:00:00</date>
    <cid>C_1239</cid>
    <source>ftp</source>
    <subcmd>moduflow-open</subcmd>
    <address>011A0A30AAA0</address>
    <result>ok</result>
  </wavenis_cmd>

  <wavenis_cmd>
    <date>2011-05-27T20:00:00</date>
    <cid>C_1240</cid>
    <source>ftp</source>
    <subcmd>raw</subcmd>
    <address>011A0A30AAA0</address>
    <result>ok</result>
    <request>10012001</request>
    <response>9001200106</response>
  </wavenis_cmd>

  <modbus_cmd>
    <date>2011-05-27T21:00:00</date>
    <cid>C_1240</cid>
    <source>ftp</source>
    <subcmd>write</subcmd>
    <result>ok</result>
  </modbus_cmd>

  <modbus>
    <date>2011-05-27T22:00:00</date>
```

```
    <module>
      <label>module1</label>
      <address>1</address>
      <ip>192.168.1.2</ip>
    </module>
    <variable>
      <name>var1</name>
      <address>2</address>
      <type>S4</type>
    </variable>
    <value>5</value>
    <status>low</status>
  </modbus>

</alarms>
```

22 ANNEXE E Schéma XSD – Supervision

supervision.xsd

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:wd="http://www.webdyn.com/WRF_supervision_20110606"
  targetNamespace="http://www.webdyn.com/WRF_supervision_20110606"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

<xsd:element name="supervision">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="uid" type="xsd:string"/>
      <xsd:choice maxOccurs="unbounded">
        <xsd:element name="status" type="wd:status"/>
        <xsd:element name="scan" type="wd:scan"/>
        <xsd:element name="timesync" type="wd:timesync"/>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:complexType name='status'>
  <xsd:sequence>
    <xsd:element name="app">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="version" type="xsd:string"/>
          <xsd:element name="kernel" type="xsd:string"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="system">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="power" type="xsd:boolean"/>
          <xsd:element name="defaults" type="xsd:string"/>
          <xsd:element name="uptime" type="xsd:string"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="com">
      <xsd:complexType>
        <xsd:sequence>
```

```
            <xsd:element name="modem">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="model" type="xsd:string"/>
                  <xsd:element name="firmware" type="xsd:string"/>
                  <xsd:element name="imei" type="xsd:string"/>
                  <xsd:element name="msisdn" type="xsd:string"/>
                  <xsd:element name="rssi" type="xsd:string"/>
                  <xsd:element name="csq" type="xsd:string"/>
                  <xsd:element name="ip" type="wd:ipv4_opt"/>
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
            <xsd:element name="ethernet">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="ip" type="wd:ipv4_opt"/>
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
            <xsd:element name="upload">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="last" type="wd:datetime_opt"/>
                  <xsd:element name="next"
type="wd:datetime_follow"/>
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="wavenis">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="address" type="wd:wavenis_address"/>
            <xsd:element name="last" type="wd:datetime_opt"/>
            <xsd:element name="modules">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="count" type="xsd:integer"/>
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="metering">
```

```
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="mbus">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="last" type="wd:datetime_opt"/>
                <xsd:element name="last_scan"
type="wd:datetime_opt"/>
                <xsd:element name="modules">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="count"
type="xsd:integer"/>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="rfid">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="count" type="xsd:integer"
minOccurs="0"/>
          <xsd:element name="tags" minOccurs="0">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="tag" type="wd:rfid_tag"
minOccurs="0"

maxOccurs="unbounded"/>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="modbus">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="next" type="wd:empty"/>
          <xsd:element name="modules">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="count" type="xsd:integer"/>
```

```
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="cid" type="xsd:string" use="optional"/>
  <xsd:attribute name="date" type="xsd:dateTime"/>
</xsd:complexType>

<xsd:complexType name='rfid_tag'>
  <xsd:sequence>
    <xsd:element name="id" type="wd:hex_string"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='scan'>
  <xsd:sequence>
    <xsd:element name="wavenis" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="modules">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="module"
type="wd:wavenis_scan_data"
                    minOccurs="0" maxOccurs="unbounded"/>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="cid" type="xsd:string" use="optional"/>
  <xsd:attribute name="date" type="xsd:dateTime"/>
</xsd:complexType>

<xsd:complexType name="wavenis_scan_data">
  <xsd:sequence>
    <xsd:element name="date" type="xsd:dateTime"/>
    <xsd:element name="address" type="wd:wavenis_address"/>
    <xsd:element name="rssi" type="xsd:integer" minOccurs="0"/>
    <xsd:element name="life-counter" type="xsd:integer"
minOccurs="0"/>
    <xsd:element name="rtc" minOccurs="0">
      <xsd:complexType>
```

```
        <xsd:sequence>
          <xsd:element name="request_date" type="xsd:dateTime"/>
          <xsd:element name="response_date" type="xsd:dateTime"/>
          <xsd:element name="date" type="xsd:dateTime"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name='timesync'>
  <xsd:sequence>
    <xsd:element name="wavenis" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="count" type="xsd:integer"/>
          <xsd:element name="skipped" minOccurs="0">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="address"
type="wd:wavenis_address"
                    minOccurs="0" maxOccurs="unbounded"/>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="cid" type="xsd:string" use="optional"/>
  <xsd:attribute name="date" type="xsd:dateTime"/>
</xsd:complexType>

<xsd:simpleType name='wavenis_address'>
  <xsd:restriction base='xsd:string'>
    <xsd:pattern value='([0-9a-zA-Z]{12})|([0-9]{5}-?[0-9]{2}-?[0-
9]{8})'/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name='ipv4'>
  <xsd:restriction base='xsd:string'>
    <xsd:pattern value='[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-
9]{1,3}'/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="empty">
```

```
  <xsd:restriction base="xsd:string">
    <xsd:length value="0"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="ipv4_opt">
  <xsd:union memberTypes="wd:ipv4 wd:empty"/>
</xsd:simpleType>

<xsd:simpleType name="datetime_opt">
  <xsd:union memberTypes="xsd:dateTime wd:empty"/>
</xsd:simpleType>

<xsd:simpleType name='follow'>
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="follow"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="datetime_follow">
  <xsd:union memberTypes="xsd:dateTime wd:follow wd:empty"/>
</xsd:simpleType>

<xsd:simpleType name='hex_string'>
        <xsd:restriction base='xsd:string'>
                <xsd:pattern value='[0-9a-zA-Z]*'/>
        </xsd:restriction>
</xsd:simpleType>

</xsd:schema>
```

23 ANNEXE F Exemple XML – Supervision

supervision.xml

```
<?xml version="1.0"?>
<supervision
  xmlns="http://www.webdyn.com/WRF_supervision_20110606"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.webdyn.com/WRF_supervision_20110606
supervision.xsd">

  <uid>07F38D</uid>

  <status cid='cmd_123' date='2011-05-26T13:12:00'>
    <app>
      <version>0.2</version>
      <kernel>2.6.35.6</kernel>
    </app>
    <system>
      <power>true</power>
      <defaults/>
      <uptime>33 days</uptime>
    </system>
    <com>
      <modem>
        <model>Cinterion BGS2-W</model>
        <firmware>11.246</firmware>
        <imei>AA-BBBBBB-CCCCCC</imei>
        <msisdn>380561234567</msisdn>
        <rssi>1</rssi>
        <csq>3</csq>
        <ip>10.0.1.23</ip>
      </modem>
      <ethernet>
        <ip>192.168.0.10</ip>
      </ethernet>
      <upload>
        <last>2011-05-26T10:30:00</last>
        <next>2011-05-27T10:30:00</next>
      </upload>
    </com>
    <wavenis>
      <address>011A1030A6E3</address>
      <last>2011-05-26T13:00:00</last>
      <modules>
        <count>58</count>
```

```
      </modules>
    </wavenis>
    <metering>
      <mbus>
        <last>2011-05-26T12:55:00</last>
        <last_scan>2011-05-21T12:05:00</last_scan>
        <modules>
          <count>8</count>
        </modules>
      </mbus>
    </metering>
    <rfid>
      <count>2</count>
      <tags>
        <tag>
          <id>23F005</id>
        </tag>
        <tag>
          <id>23F006</id>
        </tag>
      </tags>
    </rfid>
  </status>

  <scan cid='cmd_124' date='2011-05-26T13:18:00'>
    <wavenis>
      <modules>
        <module>
          <date>2011-05-26T13:18:10</date>
          <address>011A1030A5D4</address>
          <rssi>21</rssi>
          <life-counter>5389</life-counter>
          <rtc>
            <request_date>2011-05-26T13:18:11</request_date>
            <response_date>2011-05-26T13:18:14</response_date>
            <date>2011-05-26T13:13:25</date>
          </rtc>
        </module>
      </modules>
    </wavenis>
  </scan>

  <timesync cid='cmd_125' date='2011-05-26T13:27:00'>
    <wavenis>
      <count>58</count>
    </wavenis>
  </timesync>
```

```
</supervision>
```

24 ANNEXE G Schéma XSD – Données

data.xsd

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:wd="http://www.webdyn.com/WRF_data_20110606"
  targetNamespace="http://www.webdyn.com/WRF_data_20110606"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

<xsd:element name="data">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="uid" type="xsd:string"/>
      <xsd:choice maxOccurs="unbounded">
        <xsd:element name="pulse" type="wd:pulse"/>
        <xsd:element name="mbus" type="wd:mbus"/>
        <xsd:element name="wavenis" type="wd:wavenis"/>
        <xsd:element name="modbus" type="wd:modbus"/>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:complexType name='pulse'>
  <xsd:sequence>
    <xsd:element name="input_1" minOccurs="0">
      <xsd:complexType>
        <xsd:simpleContent>
          <xsd:extension base="xsd:nonNegativeInteger">
            <xsd:attribute name="label" type="xsd:string"
use="optional"/>
            <xsd:attribute name="unit" type="xsd:string"
use="optional"/>
          </xsd:extension>
        </xsd:simpleContent>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="input_2" minOccurs="0">
      <xsd:complexType>
        <xsd:simpleContent>
          <xsd:extension base="xsd:nonNegativeInteger">
            <xsd:attribute name="label" type="xsd:string"
use="optional"/>
            <xsd:attribute name="unit" type="xsd:string"
```

```
use="optional"/>
          </xsd:extension>
        </xsd:simpleContent>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="input_3" minOccurs="0">
      <xsd:complexType>
        <xsd:simpleContent>
          <xsd:extension base="xsd:nonNegativeInteger">
            <xsd:attribute name="label" type="xsd:string"
use="optional"/>
            <xsd:attribute name="unit" type="xsd:string"
use="optional"/>
          </xsd:extension>
        </xsd:simpleContent>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="date" type="xsd:dateTime"/>
</xsd:complexType>

<xsd:complexType name='mbus'>
  <xsd:sequence>
    <xsd:element name="header" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="Id" type="xsd:string"/>
          <xsd:element name="Manufacturer" type="xsd:string"
minOccurs="0"/>
          <xsd:element name="Version" type="xsd:string"
minOccurs="0"/>
          <xsd:element name="Medium" type="xsd:string"/>
          <xsd:element name="AccessNumber" type="xsd:string"/>
          <xsd:element name="Status" type="xsd:string"/>
          <xsd:element name="Signature" type="xsd:string"
minOccurs="0"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="records" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="record" minOccurs="0"
maxOccurs="unbounded">
            <xsd:complexType>
              <xsd:simpleContent>
                <xsd:extension base="xsd:string">
                    <xsd:attribute name="type">
```

```
                              <xsd:simpleType>
                                   <xsd:restriction base="xsd:string">
                                        <xsd:enumeration
value="inst"/>
                                        <xsd:enumeration value="max"/>
                                        <xsd:enumeration value="min"/>
                                        <xsd:enumeration
value="error_state"/>
                                        <xsd:enumeration
value="manufacturer"/>
                                        <xsd:enumeration
value="actual"/>
                                        <xsd:enumeration
value="stored"/>
                                   </xsd:restriction>
                              </xsd:simpleType>
                         </xsd:attribute>
                         <xsd:attribute name="unit" type="xsd:string"/>
                         <xsd:attribute name="storage"
type="xsd:nonNegativeInteger"
                              use="optional"/>
                         <xsd:attribute name="tariff"
type="xsd:nonNegativeInteger"
                              use="optional"/>
                         <xsd:attribute name="sub-unit"
type="xsd:nonNegativeInteger"
                              use="optional"/>
                    </xsd:extension>
               </xsd:simpleContent>
          </xsd:complexType>
     </xsd:element>
   </xsd:sequence>
   <xsd:attribute name="date" type="xsd:dateTime"/>
</xsd:complexType>

<xsd:complexType name='wavenis'>
   <xsd:sequence>
     <xsd:element name="header">
       <xsd:complexType>
         <xsd:sequence>
           <xsd:element name="address" type="wd:wavenis_address" />
           <xsd:element name="label" type="xsd:string" />
           <xsd:element name="type" type="wd:wavenis_type" />
           <xsd:element name="status" type="xsd:integer" />
           <xsd:element name="io_state" type="xsd:integer"
```

```
minOccurs="0" />
          <xsd:element name="retry_count" type="xsd:integer"
minOccurs="0"/>
          <xsd:element name="err_status"
type="wd:wavenis_err_status" minOccurs="0"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="records" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="record" maxOccurs="unbounded">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="date" type="xsd:dateTime" />
                <xsd:element name="input" type="wd:wavenis_input"
minOccurs="0" />
                <xsd:element name="value" type="xsd:float"
minOccurs="0" />
                <xsd:element name="event_status"
type="xsd:integer" minOccurs="0" />
                <xsd:element name="cause" type="xsd:integer"
minOccurs="0" />
                <xsd:element name="tic_fixed" type="xsd:anyType"
minOccurs="0" />
                <xsd:element name="tic" type="xsd:anyType"
minOccurs="0" />
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="date" type="xsd:dateTime" />
</xsd:complexType>

<xsd:simpleType name="wavenis_err_status">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="no_response" />
    <xsd:enumeration value="err_repeater_1" />
    <xsd:enumeration value="err_repeater_2" />
    <xsd:enumeration value="err_repeater_3" />
    <xsd:enumeration value="err_config" />
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name='wavenis_address'>
```

```
  <xsd:restriction base='xsd:string'>
    <xsd:pattern value='([0-9a-zA-Z]{12})|([0-9]{5}-?[0-9]{2}-?[0-
9]{8})' />
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="wavenis_input">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="A" />
    <xsd:enumeration value="B" />
    <xsd:enumeration value="C" />
    <xsd:enumeration value="D" />
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="wavenis_type">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="waveflow" />
    <xsd:enumeration value="wavetherm_dallas" />
    <xsd:enumeration value="wavetherm_pt100" />
    <xsd:enumeration value="wavetherm_pt1000" />
    <xsd:enumeration value="wavesense_4-20ma" />
    <xsd:enumeration value="wavesense_0-5v" />
    <xsd:enumeration value="wavelog" />
    <xsd:enumeration value="wavetic" />
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name='modbus'>
  <xsd:sequence>
    <xsd:element name="header">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="label" type="xsd:string" />
          <xsd:element name="address" type="xsd:integer" />
          <xsd:element name="ip" type="wd:ipv4" minOccurs="0" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="records" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="record" maxOccurs="unbounded">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="name" type="xsd:string" />
                <xsd:element name="address" type="wd:hex_string"
/>
```

```
                <xsd:element name="type" type="wd:modbus_type" />
                <xsd:group ref="wd:modbus_interpreted_value"
minOccurs="0"/>
                <xsd:element name="value"
type="wd:modbus_value_type"/>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="date" type="xsd:dateTime" />
</xsd:complexType>

<xsd:group name="modbus_interpreted_value">
  <xsd:sequence>
    <xsd:element name="min" type="xsd:string"/>
    <xsd:element name="max" type="xsd:string"/>
    <xsd:element name="avg" type="wd:modbus_avg_type"/>
  </xsd:sequence>
</xsd:group>

<xsd:simpleType name='modbus_type'>
        <xsd:restriction base='xsd:string'>
                <xsd:pattern value='S[0,1,3,4]'/>
        </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="modbus_avg_type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="count" type="xsd:integer"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="modbus_value_type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="date" type="xsd:dateTime"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:simpleType name='ipv4'>
        <xsd:restriction base='xsd:string'>
                <xsd:pattern value='[0-9]{1,3}.[0-9]{1,3}.[0-
```

```
9]{1,3}.[0-9]{1,3}'/>
        </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name='hex_string'>
        <xsd:restriction base='xsd:string'>
                <xsd:pattern value='0x[0-9a-zA-Z]*'/>
        </xsd:restriction>
</xsd:simpleType>

</xsd:schema>
```

25 ANNEXE H Format CSV – Données

Le format CSV (Comma Separated Values) est un format sans définition formelle.

Néanmoins, il suit les règles suivantes :

- Une ligne contient un seul enregistrement
- Chaque enregistrement correspond à une seule ligne
- Chaque ligne se termine par un retour de ligne.
- Chaque ligne contient le même nombre de champs.
- Chaque champ est séparé par une virgule.

Chaque ligne est formatée comme suit:

<timestamp>,<Source>,<Address>,,,<Label>,<Input>,<Valeur>

Avec:

- Horodatage: JJ/MM/AAAA HH:MM
- Source : Les valeurs possibles sont:
  - FL: WaveFlow
  - TH: WaveTherm
  - LO: WaveLog
  - SE: WaveSense
- Adresse : adresse Wavenis
- Label : module de Wavenis s'il est configuré, vide sinon.
- Entrée: identifiant d'entrée (A, B, C ou D).
- Valeur: Valeur de l'entrée donnée.

26 ANNEXE I Schéma XSD – Commandes

command.xsd

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">



  <head>
    <title>
      command.xsd in
Projets/Clients/WEBDYN/Gateway/WG310P/trunk/XML_Schemas
     – webdyn
    </title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"
/>
    <!--[if IE]><script type="text/javascript">window.location.hash =
window.location.hash;</script><![endif]-->
        <link rel="search" href="/trac/search" />
        <link rel="help" href="/trac/wiki/TracGuide" />
        <link rel="alternate"
href="/trac/browser/Projets/Clients/WEBDYN/Gateway/WG310P/trunk/XML_Sch
emas/command.xsd?format=txt" type="text/plain" title="Plain Text"
/><link rel="alternate"
href="/trac/export/3915/Projets/Clients/WEBDYN/Gateway/WG310P/trunk/XML
_Schemas/command.xsd" type="application/xml; charset=utf-8"
title="Original Format" />
        <link rel="start" href="/trac/wiki" />
        <link rel="stylesheet" href="/trac/chrome/common/css/trac.css"
type="text/css" /><link rel="stylesheet"
href="/trac/chrome/common/css/code.css" type="text/css" /><link
rel="stylesheet" href="/trac/pygments/trac.css" type="text/css" /><link
rel="stylesheet" href="/trac/chrome/common/css/browser.css"
type="text/css" />
        <link rel="prev"
href="/trac/browser/Projets/Clients/WEBDYN/Gateway/WG310P/trunk/XML_Sch
emas/command.xsd?rev=3715" title="Revision 3715" />
        <link rel="shortcut icon" href="/trac/chrome/common/trac.ico"
type="image/x-icon" />
        <link rel="icon" href="/trac/chrome/common/trac.ico"
type="image/x-icon" />
      <link type="application/opensearchdescription+xml" rel="search"
```

```
href="/trac/search/opensearch" title="Search webdyn" />
    <script type="text/javascript"
src="/trac/chrome/common/js/jquery.js"></script><script
type="text/javascript"
src="/trac/chrome/common/js/babel.js"></script><script
type="text/javascript"
src="/trac/chrome/common/js/trac.js"></script><script
type="text/javascript"
src="/trac/chrome/common/js/search.js"></script><script
type="text/javascript" src="/trac/chrome/site/md5.js"></script>
    <!--[if lt IE 7]>
    <script type="text/javascript"
src="/trac/chrome/common/js/ie_pre7_hacks.js"></script>
    <![endif]-->
    <script type="text/javascript"
src="/trac/chrome/common/js/folding.js"></script>
    <script type="text/javascript">
      jQuery(document).ready(function($) {
        $(".trac-toggledeleted").show().click(function() {
                  $(this).siblings().find(".trac-deleted").toggle();
                  return false;
        }).click();
        $("#jumploc input").hide();
        $("#jumploc select").change(function () {
          this.parentNode.parentNode.submit();
        });
          $('#preview table.code').enableCollapsibleColumns($('#preview
table.code thead th.content'));
      });
    </script>
  </head>
  <body>
    <div id="banner">
      <div id="header">
        <a id="logo" href="/trac/wiki/TracIni#header_logo-section"><img
src="/trac/chrome/site/webdyn.jpg" alt="(please configure the
[header_logo] section in trac.ini)" /></a>
      </div>
      <form id="search" action="/trac/search" method="get">
        <div>
          <label for="proj-search">Search:</label>
          <input type="text" id="proj-search" name="q" size="18"
value="" />
          <input type="submit" value="Search" />
        </div>
      </form>
      <div id="metanav" class="nav">
    <ul>
```

```
      <li class="first"><a href="/trac/login">Login</a></li><li><a
href="/trac/prefs">Preferences</a></li><li><a
href="/trac/wiki/TracGuide">Help/Guide</a></li><li><a
href="/trac/about">About Trac</a></li><li class="last"><a
href="/trac/register">Register</a></li>
    </ul>
  </div>
    </div>
    <div id="mainnav" class="nav">
    <ul>
      <li class="first"><a href="/trac/wiki">Wiki</a></li><li><a
href="/trac/timeline">Timeline</a></li><li><a
href="/trac/roadmap">Roadmap</a></li><li class="active"><a
href="/trac/browser">Browse Source</a></li><li><a
href="/trac/report">View Tickets</a></li><li><a
href="/trac/search">Search</a></li><li><a
href="/trac/ticketcalendar">Calendar</a></li><li><a
href="/trac/pdashboard">Metrics</a></li><li class="last"><a
href="/trac/ticketgantt">Gantt Ticket</a></li>
    </ul>
  </div>
    <div id="main">
      <div id="ctxtnav" class="nav">
        <h2>Context Navigation</h2>
          <ul>
              <li class="first"><span>&larr; <a class="prev"
href="/trac/browser/Projets/Clients/WEBDYN/Gateway/WG310P/trunk/XML_Sch
emas/command.xsd?rev=3715" title="Revision 3715">Previous
Revision</a></span></li><li><span class="missing">Next Revision
&rarr;</span></li><li><a
href="/trac/browser/Projets/Clients/WEBDYN/Gateway/WG310P/trunk/XML_Sch
emas/command.xsd?annotate=blame" title="Annotate each line with the
last changed revision (this can be time
consuming...)">Blame</a></li><li class="last"><a
href="/trac/log/Projets/Clients/WEBDYN/Gateway/WG310P/trunk/XML_Schemas
/command.xsd">Revision Log</a></li>
          </ul>
        <hr />
      </div>
    <div id="content" class="browser">
          <h1>
<a class="pathentry first" href="/trac/browser?order=name" title="Go to
repository index">source:</a>
<a class="pathentry" href="/trac/browser/Projets?order=name"
title="View Projets">Projets</a><span class="pathentry sep">/</span><a
class="pathentry" href="/trac/browser/Projets/Clients?order=name"
title="View Clients">Clients</a><span class="pathentry sep">/</span><a
class="pathentry"
```

```
href="/trac/browser/Projets/Clients/WEBDYN?order=name" title="View
WEBDYN">WEBDYN</a><span class="pathentry sep">/</span><a
class="pathentry"
href="/trac/browser/Projets/Clients/WEBDYN/Gateway?order=name"
title="View Gateway">Gateway</a><span class="pathentry sep">/</span><a
class="pathentry"
href="/trac/browser/Projets/Clients/WEBDYN/Gateway/WG310P?order=name"
title="View WG310P">WG310P</a><span class="pathentry sep">/</span><a
class="pathentry"
href="/trac/browser/Projets/Clients/WEBDYN/Gateway/WG310P/trunk?order=n
ame" title="View trunk">trunk</a><span class="pathentry sep">/</span><a
class="pathentry"
href="/trac/browser/Projets/Clients/WEBDYN/Gateway/WG310P/trunk/XML_Sch
emas?order=name" title="View XML_Schemas">XML_Schemas</a><span
class="pathentry sep">/</span><a class="pathentry"
href="/trac/browser/Projets/Clients/WEBDYN/Gateway/WG310P/trunk/XML_Sch
emas/command.xsd?order=name" title="View command.xsd">command.xsd</a>
<span class="pathentry sep">@</span>
  <a class="pathentry" href="/trac/changeset/3915/Projets" title="View
changeset 3915">3915</a>
<br style="clear: both" />
</h1>
        <div id="jumprev">
          <form action="" method="get">
            <div>
              <label for="rev">
                View revision:</label>
              <input type="text" id="rev" name="rev" size="6" />
            </div>
          </form>
        </div>
      <table id="info" summary="Revision info">
        <tr>
          <th scope="col">Revision <a
href="/trac/changeset/3716/Projets">3716</a>,
            <span title="4047 bytes">4.0 KB</span>
            checked in by ghofmann, <a class="timeline"
href="/trac/timeline?from=2012-11-
19T18%3A15%3A52%2B01%3A00&amp;precision=second" title="2012-11-
19T18:15:52+01:00 in Timeline">3 weeks</a> ago
            (<a
href="/trac/changeset/3716/Projets/Clients/WEBDYN/Gateway/WG310P/trunk/
XML_Schemas/command.xsd">diff</a>)</th>
        </tr>
        <tr>
          <td class="message searchable">
            <p>
Fix modbus command syntax which do not accept rtu addresses.<br />
```

```
</p>
          </td>
        </tr>
      </table>
      <div id="preview" class="searchable">

  <table class="code"><thead><tr><th class="lineno" title="Line
numbers">Line</th><th class="content"> </th></tr></thead><tbody><tr><th
id="L1"><a href="#L1">1</a></th><td><span class="cp">&lt;?xml
version="1.0"?&gt;</span></td></tr><tr><th id="L2"><a
href="#L2">2</a></th><td><span class="nt">&lt;xsd:schema</span> <span
class="na">xmlns:xsd=</span><span
class="s">"http://www.w3.org/2001/XMLSchema"</span></td></tr><tr><th
id="L3"><a href="#L3">3</a></th><td>        <span
class="na">xmlns:wd=</span><span
class="s">"http://www.webdyn.com/WRF_command_20110606"</span></td></tr>
<tr><th id="L4"><a href="#L4">4</a></th><td>        <span
class="na">targetNamespace=</span><span
class="s">"http://www.webdyn.com/WRF_command_20110606"</span></td></tr>
<tr><th id="L5"><a href="#L5">5</a></th><td>        <span
class="na">elementFormDefault=</span><span
class="s">"qualified"</span></td></tr><tr><th id="L6"><a
href="#L6">6</a></th><td>        <span
class="na">attributeFormDefault=</span><span
class="s">"unqualified"</span><span
class="nt">&gt;</span></td></tr><tr><th id="L7"><a
href="#L7">7</a></th><td></td></tr><tr><th id="L8"><a
href="#L8">8</a></th><td><span class="nt">&lt;xsd:element</span> <span
class="na">name=</span><span class="s">"commands"</span><span
class="nt">&gt;</span></td></tr><tr><th id="L9"><a
href="#L9">9</a></th><td>        <span
class="nt">&lt;xsd:complexType&gt;</span></td></tr><tr><th id="L10"><a
href="#L10">10</a></th><td>                <span
class="nt">&lt;xsd:sequence&gt;</span></td></tr><tr><th id="L11"><a
href="#L11">11</a></th><td>                        <span
class="nt">&lt;xsd:element</span> <span class="na">name=</span><span
class="s">"cmd"</span> <span class="na">maxOccurs=</span><span
class="s">"unbounded"</span><span
class="nt">&gt;</span></td></tr><tr><th id="L12"><a
href="#L12">12</a></th><td>                                <span
class="nt">&lt;xsd:complexType&gt;</span></td></tr><tr><th id="L13"><a
href="#L13">13</a></th><td>                                        <spa
n class="nt">&lt;xsd:sequence&gt;</span></td></tr><tr><th id="L14"><a
href="#L14">14</a></th><td>
    <span class="nt">&lt;xsd:choice&gt;</span></td></tr><tr><th
id="L15"><a
href="#L15">15</a></th><td>
            <span class="nt">&lt;xsd:element</span> <span
```

```
class="na">name=</span><span class="s">"reboot"</span> <span
class="na">type=</span><span class="s">"wd:cmd_simple"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L16"><a
href="#L16">16</a></th><td>
            <span class="nt">&lt;xsd:element</span> <span
class="na">name=</span><span class="s">"factory"</span> <span
class="na">type=</span><span class="s">"wd:cmd_simple"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L17"><a
href="#L17">17</a></th><td>
            <span class="nt">&lt;xsd:element</span> <span
class="na">name=</span><span class="s">"update"</span> <span
class="na">type=</span><span class="s">"wd:cmd_update"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L18"><a
href="#L18">18</a></th><td>
            <span class="nt">&lt;xsd:element</span> <span
class="na">name=</span><span class="s">"scan"</span> <span
class="na">type=</span><span class="s">"wd:cmd_scan"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L19"><a
href="#L19">19</a></th><td>
            <span class="nt">&lt;xsd:element</span> <span
class="na">name=</span><span class="s">"timesync"</span> <span
class="na">type=</span><span class="s">"wd:cmd_timesync"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L20"><a
href="#L20">20</a></th><td>
            <span class="nt">&lt;xsd:element</span> <span
class="na">name=</span><span class="s">"wavenis"</span> <span
class="na">type=</span><span class="s">"wd:cmd_wavenis"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L21"><a
href="#L21">21</a></th><td>
            <span class="nt">&lt;xsd:element</span> <span
class="na">name=</span><span class="s">"modbus"</span> <span
class="na">type=</span><span class="s">"wd:cmd_modbus"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L22"><a
href="#L22">22</a></th><td>
            <span class="nt">&lt;xsd:element</span> <span
class="na">name=</span><span class="s">"status"</span> <span
class="na">type=</span><span class="s">"wd:cmd_simple"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L23"><a
href="#L23">23</a></th><td>
            <span class="nt">&lt;xsd:element</span> <span
class="na">name=</span><span class="s">"log"</span> <span
class="na">type=</span><span class="s">"wd:cmd_simple"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L24"><a
href="#L24">24</a></th><td>
            <span class="nt">&lt;xsd:element</span> <span
class="na">name=</span><span class="s">"d_output"</span> <span
class="na">type=</span><span class="s">"wd:cmd_d_output"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L25"><a
```

```
href="#L25">25</a></th><td>
    <span class="nt">&lt;/xsd:choice&gt;</span></td></tr><tr><th
id="L26"><a
href="#L26">26</a></th><td>                                                    <spa
n class="nt">&lt;/xsd:sequence&gt;</span></td></tr><tr><th id="L27"><a
href="#L27">27</a></th><td>                                                    <spa
n class="nt">&lt;xsd:attribute</span> <span
class="na">name=</span><span class="s">"uid"</span> <span
class="na">type=</span><span class="s">"xsd:string"</span> <span
class="na">use=</span><span class="s">"optional"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L28"><a
href="#L28">28</a></th><td>                                                    <spa
n class="nt">&lt;xsd:attribute</span> <span
class="na">name=</span><span class="s">"cid"</span> <span
class="na">type=</span><span class="s">"xsd:string"</span> <span
class="na">use=</span><span class="s">"optional"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L29"><a
href="#L29">29</a></th><td>                                          <span
class="nt">&lt;/xsd:complexType&gt;</span></td></tr><tr><th id="L30"><a
href="#L30">30</a></th><td>                                <span
class="nt">&lt;/xsd:element&gt;</span></td></tr><tr><th id="L31"><a
href="#L31">31</a></th><td>                        <span
class="nt">&lt;/xsd:sequence&gt;</span></td></tr><tr><th id="L32"><a
href="#L32">32</a></th><td>        <span
class="nt">&lt;/xsd:complexType&gt;</span></td></tr><tr><th id="L33"><a
href="#L33">33</a></th><td><span
class="nt">&lt;/xsd:element&gt;</span></td></tr><tr><th id="L34"><a
href="#L34">34</a></th><td></td></tr><tr><th id="L35"><a
href="#L35">35</a></th><td><span
class="nt">&lt;xsd:complexType</span> <span
class="na">name=</span><span class="s">"cmd_simple"</span><span
class="nt">&gt;</span></td></tr><tr><th id="L36"><a
href="#L36">36</a></th><td>        <span
class="nt">&lt;xsd:complexContent&gt;</span></td></tr><tr><th
id="L37"><a href="#L37">37</a></th><td>                <span
class="nt">&lt;xsd:restriction</span> <span
class="na">base=</span><span class="s">"xsd:anyType"</span><span
class="nt">&gt;</span></td></tr><tr><th id="L38"><a
href="#L38">38</a></th><td>                        <span
class="nt">&lt;/xsd:restriction&gt;</span></td></tr><tr><th id="L39"><a
href="#L39">39</a></th><td>        <span
class="nt">&lt;/xsd:complexContent&gt;</span></td></tr><tr><th
id="L40"><a href="#L40">40</a></th><td><span
class="nt">&lt;/xsd:complexType&gt;</span></td></tr><tr><th id="L41"><a
href="#L41">41</a></th><td></td></tr><tr><th id="L42"><a
href="#L42">42</a></th><td><span
class="nt">&lt;xsd:complexType</span> <span
class="na">name=</span><span class="s">"cmd_update"</span><span
```

```
class="nt">&gt;</span></td></tr><tr><th id="L43"><a
href="#L43">43</a></th><td>        <span
class="nt">&lt;xsd:sequence&gt;</span></td></tr><tr><th id="L44"><a
href="#L44">44</a></th><td>                <span
class="nt">&lt;xsd:element</span> <span class="na">name=</span><span
class="s">"firmware"</span> <span class="na">type=</span><span
class="s">"xsd:string"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L45"><a
href="#L45">45</a></th><td>                <span
class="nt">&lt;xsd:element</span> <span class="na">name=</span><span
class="s">"checksum"</span> <span class="na">type=</span><span
class="s">"xsd:string"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L46"><a
href="#L46">46</a></th><td>        <span
class="nt">&lt;/xsd:sequence&gt;</span></td></tr><tr><th id="L47"><a
href="#L47">47</a></th><td><span
class="nt">&lt;/xsd:complexType&gt;</span></td></tr><tr><th id="L48"><a
href="#L48">48</a></th><td></td></tr><tr><th id="L49"><a
href="#L49">49</a></th><td><span
class="nt">&lt;xsd:complexType</span> <span
class="na">name=</span><span class="s">"cmd_scan"</span><span
class="nt">&gt;</span></td></tr><tr><th id="L50"><a
href="#L50">50</a></th><td>        <span
class="nt">&lt;xsd:sequence&gt;</span></td></tr><tr><th id="L51"><a
href="#L51">51</a></th><td>                <span
class="nt">&lt;xsd:element</span> <span class="na">name=</span><span
class="s">"address"</span> <span class="na">type=</span><span
class="s">"wd:wavenis_address"</span></td></tr><tr><th id="L52"><a
href="#L52">52</a></th><td>                        <span
class="na">minOccurs=</span><span class="s">"0"</span> <span
class="na">maxOccurs=</span><span class="s">"unbounded"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L53"><a
href="#L53">53</a></th><td>        <span
class="nt">&lt;/xsd:sequence&gt;</span></td></tr><tr><th id="L54"><a
href="#L54">54</a></th><td>        <span
class="nt">&lt;xsd:attribute</span> <span class="na">name=</span><span
class="s">"mode"</span><span class="nt">&gt;</span></td></tr><tr><th
id="L55"><a href="#L55">55</a></th><td>                <span
class="nt">&lt;xsd:simpleType&gt;</span></td></tr><tr><th id="L56"><a
href="#L56">56</a></th><td>                        <span
class="nt">&lt;xsd:list</span> <span class="na">itemType=</span><span
class="s">"wd:scan_modes"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L57"><a
href="#L57">57</a></th><td>                <span
class="nt">&lt;/xsd:simpleType&gt;</span></td></tr><tr><th id="L58"><a
href="#L58">58</a></th><td>        <span
class="nt">&lt;/xsd:attribute&gt;</span></td></tr><tr><th id="L59"><a
href="#L59">59</a></th><td><span
```

```
class="nt">&lt;/xsd:complexType&gt;</span></td></tr><tr><th id="L60"><a
href="#L60">60</a></th><td></td></tr><tr><th id="L61"><a
href="#L61">61</a></th><td><span
class="nt">&lt;xsd:simpleType</span> <span class="na">name=</span><span
class="s">"scan_modes"</span><span
class="nt">&gt;</span></td></tr><tr><th id="L62"><a
href="#L62">62</a></th><td>        <span
class="nt">&lt;xsd:restriction</span> <span
class="na">base=</span><span class="s">"xsd:string"</span><span
class="nt">&gt;</span></td></tr><tr><th id="L63"><a
href="#L63">63</a></th><td>                <span
class="nt">&lt;xsd:enumeration</span> <span
class="na">value=</span><span class="s">"data"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L64"><a
href="#L64">64</a></th><td>                <span
class="nt">&lt;xsd:enumeration</span> <span
class="na">value=</span><span class="s">"rssi"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L65"><a
href="#L65">65</a></th><td>                <span
class="nt">&lt;xsd:enumeration</span> <span
class="na">value=</span><span class="s">"life-counter"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L66"><a
href="#L66">66</a></th><td>                <span
class="nt">&lt;xsd:enumeration</span> <span
class="na">value=</span><span class="s">"rtc"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L67"><a
href="#L67">67</a></th><td>                <span
class="nt">&lt;xsd:enumeration</span> <span
class="na">value=</span><span class="s">"tic"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L68"><a
href="#L68">68</a></th><td>        <span
class="nt">&lt;/xsd:restriction&gt;</span></td></tr><tr><th id="L69"><a
href="#L69">69</a></th><td><span
class="nt">&lt;/xsd:simpleType&gt;</span>   </td></tr><tr><th
id="L70"><a href="#L70">70</a></th><td></td></tr><tr><th id="L71"><a
href="#L71">71</a></th><td><span
class="nt">&lt;xsd:complexType</span> <span
class="na">name=</span><span class="s">"cmd_timesync"</span><span
class="nt">&gt;</span></td></tr><tr><th id="L72"><a
href="#L72">72</a></th><td>        <span
class="nt">&lt;xsd:sequence&gt;</span></td></tr><tr><th id="L73"><a
href="#L73">73</a></th><td>                <span
class="nt">&lt;xsd:element</span> <span class="na">name=</span><span
class="s">"address"</span> <span class="na">type=</span><span
class="s">"wd:wavenis_address"</span></td></tr><tr><th id="L74"><a
href="#L74">74</a></th><td>                        <span
class="na">minOccurs=</span><span class="s">"0"</span> <span
class="na">maxOccurs=</span><span class="s">"unbounded"</span><span
```

```
class="nt">/&gt;</span></td></tr><tr><th id="L75"><a
href="#L75">75</a></th><td>        <span
class="nt">&lt;/xsd:sequence&gt;</span></td></tr><tr><th id="L76"><a
href="#L76">76</a></th><td><span
class="nt">&lt;/xsd:complexType&gt;</span></td></tr><tr><th id="L77"><a
href="#L77">77</a></th><td></td></tr><tr><th id="L78"><a
href="#L78">78</a></th><td><span
class="nt">&lt;xsd:complexType</span> <span
class="na">name=</span><span class="s">"cmd_wavenis"</span><span
class="nt">&gt;</span></td></tr><tr><th id="L79"><a
href="#L79">79</a></th><td>        <span
class="nt">&lt;xsd:sequence&gt;</span></td></tr><tr><th id="L80"><a
href="#L80">80</a></th><td>                <span
class="nt">&lt;xsd:element</span> <span class="na">name=</span><span
class="s">"address"</span> <span class="na">type=</span><span
class="s">"wd:wavenis_address"</span></td></tr><tr><th id="L81"><a
href="#L81">81</a></th><td>                        <span
class="na">minOccurs=</span><span class="s">"0"</span> <span
class="na">maxOccurs=</span><span class="s">"unbounded"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L82"><a
href="#L82">82</a></th><td>        <span
class="nt">&lt;/xsd:sequence&gt;</span></td></tr><tr><th id="L83"><a
href="#L83">83</a></th><td>        <span
class="nt">&lt;xsd:attribute</span> <span class="na">name=</span><span
class="s">"subcmd"</span> <span class="na">type=</span><span
class="s">"xsd:string"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L84"><a
href="#L84">84</a></th><td>        <span
class="nt">&lt;xsd:attribute</span> <span class="na">name=</span><span
class="s">"data"</span> <span class="na">type=</span><span
class="s">"wd:hex_string"</span> <span class="na">use=</span><span
class="s">"optional"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L85"><a
href="#L85">85</a></th><td><span
class="nt">&lt;/xsd:complexType&gt;</span></td></tr><tr><th id="L86"><a
href="#L86">86</a></th><td></td></tr><tr><th id="L87"><a
href="#L87">87</a></th><td><span
class="nt">&lt;xsd:complexType</span> <span
class="na">name=</span><span class="s">"cmd_d_output"</span><span
class="nt">&gt;</span></td></tr><tr><th id="L88"><a
href="#L88">88</a></th><td>        <span
class="nt">&lt;xsd:attribute</span> <span class="na">name=</span><span
class="s">"subcmd"</span> <span class="na">type=</span><span
class="s">"wd:d_output_cmds"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L89"><a
href="#L89">89</a></th><td><span
class="nt">&lt;/xsd:complexType&gt;</span></td></tr><tr><th id="L90"><a
href="#L90">90</a></th><td></td></tr><tr><th id="L91"><a
```

```
href="#L91">91</a></th><td><span
class="nt">&lt;xsd:simpleType</span> <span class="na">name=</span><span
class="s">"d_output_cmds"</span><span
class="nt">&gt;</span></td></tr><tr><th id="L92"><a
href="#L92">92</a></th><td>        <span
class="nt">&lt;xsd:restriction</span> <span
class="na">base=</span><span class="s">"xsd:string"</span><span
class="nt">&gt;</span></td></tr><tr><th id="L93"><a
href="#L93">93</a></th><td>                <span
class="nt">&lt;xsd:enumeration</span> <span
class="na">value=</span><span class="s">"open"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L94"><a
href="#L94">94</a></th><td>                <span
class="nt">&lt;xsd:enumeration</span> <span
class="na">value=</span><span class="s">"close"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L95"><a
href="#L95">95</a></th><td>        <span
class="nt">&lt;/xsd:restriction&gt;</span></td></tr><tr><th id="L96"><a
href="#L96">96</a></th><td><span
class="nt">&lt;/xsd:simpleType&gt;</span></td></tr><tr><th id="L97"><a
href="#L97">97</a></th><td></td></tr><tr><th id="L98"><a
href="#L98">98</a></th><td><span
class="nt">&lt;xsd:complexType</span> <span
class="na">name=</span><span class="s">"cmd_modbus"</span><span
class="nt">&gt;</span></td></tr><tr><th id="L99"><a
href="#L99">99</a></th><td>        <span
class="nt">&lt;xsd:sequence&gt;</span></td></tr><tr><th id="L100"><a
href="#L100">100</a></th><td>                <span
class="nt">&lt;xsd:element</span> <span class="na">name=</span><span
class="s">"address"</span> <span class="na">type=</span><span
class="s">"wd:modbus_cmd_string"</span> <span
class="na">maxOccurs=</span><span class="s">"unbounded"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L101"><a
href="#L101">101</a></th><td>        <span
class="nt">&lt;/xsd:sequence&gt;</span></td></tr><tr><th id="L102"><a
href="#L102">102</a></th><td>        <span
class="nt">&lt;xsd:attribute</span> <span class="na">name=</span><span
class="s">"subcmd"</span> <span class="na">type=</span><span
class="s">"xsd:string"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L103"><a
href="#L103">103</a></th><td>        <span
class="nt">&lt;xsd:attribute</span> <span class="na">name=</span><span
class="s">"data"</span> <span class="na">type=</span><span
class="s">"wd:hex_string"</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L104"><a
href="#L104">104</a></th><td><span
class="nt">&lt;/xsd:complexType&gt;</span></td></tr><tr><th
id="L105"><a href="#L105">105</a></th><td></td></tr><tr><th
```

```
id="L106"><a href="#L106">106</a></th><td><span
class="nt">&lt;xsd:simpleType</span> <span class="na">name=</span><span
class="s">'modbus_cmd_string'</span><span
class="nt">&gt;</span></td></tr><tr><th id="L107"><a
href="#L107">107</a></th><td>        <span
class="nt">&lt;xsd:restriction</span> <span
class="na">base=</span><span class="s">'xsd:string'</span><span
class="nt">&gt;</span></td></tr><tr><th id="L108"><a
href="#L108">108</a></th><td>                <span
class="nt">&lt;xsd:pattern</span> <span class="na">value=</span><span
class="s">'([0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}:)?[0-
9]{1,4}/S[0,1,3,4]@0x[0-9a-zA-Z]{1,4}'</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L109"><a
href="#L109">109</a></th><td>        <span
class="nt">&lt;/xsd:restriction&gt;</span></td></tr><tr><th
id="L110"><a href="#L110">110</a></th><td><span
class="nt">&lt;/xsd:simpleType&gt;</span></td></tr><tr><th id="L111"><a
href="#L111">111</a></th><td></td></tr><tr><th id="L112"><a
href="#L112">112</a></th><td><span
class="nt">&lt;xsd:simpleType</span> <span class="na">name=</span><span
class="s">'hex_string'</span><span
class="nt">&gt;</span></td></tr><tr><th id="L113"><a
href="#L113">113</a></th><td>        <span
class="nt">&lt;xsd:restriction</span> <span
class="na">base=</span><span class="s">'xsd:string'</span><span
class="nt">&gt;</span></td></tr><tr><th id="L114"><a
href="#L114">114</a></th><td>                <span
class="nt">&lt;xsd:pattern</span> <span class="na">value=</span><span
class="s">'[0-9a-zA-Z]*'</span><span
class="nt">/&gt;</span></td></tr><tr><th id="L115"><a
href="#L115">115</a></th><td>        <span
class="nt">&lt;/xsd:restriction&gt;</span></td></tr><tr><th
id="L116"><a href="#L116">116</a></th><td><span
class="nt">&lt;/xsd:simpleType&gt;</span></td></tr><tr><th id="L117"><a
href="#L117">117</a></th><td></td></tr><tr><th id="L118"><a
href="#L118">118</a></th><td><span
class="nt">&lt;xsd:simpleType</span> <span class="na">name=</span><span
class="s">'wavenis_address'</span><span
class="nt">&gt;</span></td></tr><tr><th id="L119"><a
href="#L119">119</a></th><td>        <span
class="nt">&lt;xsd:restriction</span> <span
class="na">base=</span><span class="s">'xsd:string'</span><span
class="nt">&gt;</span></td></tr><tr><th id="L120"><a
href="#L120">120</a></th><td>                <span
class="nt">&lt;xsd:pattern</span> <span class="na">value=</span><span
class="s">'([0-9a-zA-Z]{12})|([0-9]{5}-?[0-9]{2}-?[0-
9]{8})'</span><span class="nt">/&gt;</span></td></tr><tr><th
id="L121"><a href="#L121">121</a></th><td>        <span
```

```
class="nt">&lt;/xsd:restriction&gt;</span></td></tr><tr><th
id="L122"><a href="#L122">122</a></th><td><span
class="nt">&lt;/xsd:simpleType&gt;</span></td></tr><tr><th id="L123"><a
href="#L123">123</a></th><td></td></tr><tr><th id="L124"><a
href="#L124">124</a></th><td><span
class="nt">&lt;/xsd:schema&gt;</span></td></tr></tbody></table>

      </div>
      <div id="help"><strong>Note:</strong> See <a
href="/trac/wiki/TracBrowser">TracBrowser</a>
        for help on using the repository browser.</div>
      <div id="anydiff">
        <form action="/trac/diff" method="get">
          <div class="buttons">
            <input type="hidden" name="new_path"
value="/Projets/Clients/WEBDYN/Gateway/WG310P/trunk/XML_Schemas/command
.xsd" />
            <input type="hidden" name="old_path"
value="/Projets/Clients/WEBDYN/Gateway/WG310P/trunk/XML_Schemas/command
.xsd" />
            <input type="hidden" name="new_rev" />
            <input type="hidden" name="old_rev" />
            <input type="submit" value="View changes..." title="Select
paths and revs for Diff" />
          </div>
        </form>
      </div>
    </div>
    <div id="altlinks">
      <h3>Download in other formats:</h3>
      <ul>
        <li class="first">
          <a rel="nofollow"
href="/trac/browser/Projets/Clients/WEBDYN/Gateway/WG310P/trunk/XML_Sch
emas/command.xsd?format=txt">Plain Text</a>
        </li><li class="last">
          <a rel="nofollow"
href="/trac/export/3915/Projets/Clients/WEBDYN/Gateway/WG310P/trunk/XML
_Schemas/command.xsd">Original Format</a>
        </li>
      </ul>
    </div>
    </div>
    <div id="footer" lang="en" xml:lang="en"><hr />
      <a id="tracpowered" href="http://trac.edgewall.org/"><img
src="/trac/chrome/common/trac_logo_mini.png" height="30" width="107"
alt="Trac Powered" /></a>
      <p class="left">Powered by <a href="/trac/about"><strong>Trac
```

```
0.12.3</strong></a><br />
        By <a href="http://www.edgewall.org/">Edgewall
Software</a>.</p>
      <p class="right">Visit the Trac open source project at<br /><a
href="http://trac.edgewall.org/">http://trac.edgewall.org/</a></p>
    </div>
  </body>
</html>
```

27 ANNEXE J Exemple XML – Commandes

command.xml

```
<?xml version="1.0"?>
<commands
  xmlns="http://www.webdyn.com/WRF_command_20110606"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.webdyn.com/WRF_command_20110606
command.xsd">

<cmd cid="C_1234">
  <timesync>
    <address>011A0A30AAA0</address>
    <address>011A0A30AAA1</address>
  </timesync>
</cmd>

<cmd cid="C_1235">
  <scan mode="rtc life-counter">
    <address>011A0A30AAA0</address>
    <address>011A0A30AAA1</address>
  </scan>
</cmd>

<cmd cid="C_1236">
  <scan mode="data"/>
</cmd>

<cmd cid="C_1237">
   <status/>
</cmd>

<cmd cid="C_1238">
  <update>
    <firmware>wrf_wavenis_v101.bin</firmware>
    <checksum>c1fb7d81f3d53a8b7bf94098115249d3</checksum>
  </update>
</cmd>

<cmd cid="C_1239">
   <wavenis subcmd="moduflow-open">
    <address>011A0A30AAA0</address>
  </wavenis>
</cmd>
```

```
<cmd cid="C_1240">
  <wavenis subcmd="raw" data="10012001">
    <address>011A0A30AAA0</address>
  </wavenis>
</cmd>

<cmd cid="C_1241">
  <d_output subcmd="open"/>
</cmd>

<cmd cid="C_1242">
  <d_output subcmd="close"/>
</cmd>

<cmd cid="C_1243">
  <modbus subcmd="write">
    <address>192.168.0.17:223/S3@0x0F56</address>
  </modbus>
</cmd>

<cmd cid="C_1243">
  <modbus subcmd="write">
    <address>223/S3@0x0F56</address>
  </modbus>
</cmd>

</commands>
```